Cat.No. V058–E1–1

NT4S/NT15S/NT18S

Programmable Terminal

MULTI-VENDOR CONNECTIONS

OMRON

NT4S/NT15S/NT18S Programmable Terminal

Multi-vendor connections

Produced March 1999

OMRON Product References

All OMRON products are capitalized in this manual. The word 'Unit' is also capitalized when it refers to an OMRON product, regardless of whether or not it appears in the proper name of the product.

The abbreviation 'Ch,' which appears in some displays and on some OMRON products, often means 'word' and is abbreviated 'Wd' in documentation in this sense.

The abbreviation 'PC' means Programmable Controller and is not used as an abbreviation for anything else.

The abbreviation 'Host' means a controller such as an FA computer which controls a PT (programmable terminal).

Visual Aids

The following headings appear in the left column of the manual to help you locate different types of information.

Note Indicates information of particular interest for efficient and convenient operation of the product.

1, 2, 3... 1. Indicates lists of one sort or another, such as procedures, checklists, etc.

TABLE OF CONTENTS

About this Manual:

This manual describes how to connect the NT4S, NT15S and NT18S to non Omron PLC's.

Please read this manual carefully and be sure you understand the information provided before attempting to install and operate the NT-series programmable terminal NT4S/NT15S/NT18S.

Related Manuals and Their Contents:

The related manuals are listed below.

The □ symbol at the end of the manual number is the revision history number.

[Connecting the NT4S/NT15S/NT18S to PLC's other than Omron.]

- NT4S/NT15S/NT18S Programmable terminal, multi-vendor connections (V058-E1-□) . . . . . This manual

  The NT4S/NT15S/NT18S can also be connected to other PLC's then Omron only. This manual describes how to connect to other PLC's.

[Operating the programmable terminal and communicating with the host]

- NT4S/NT15S/NT18S Programmable Terminal Operation Manual (V056-E1-□)

This operation manual is the manual for the NT4S/NT15S/NT18S itself.

This operation manual describes the functions and handling of both the programmable terminal body and the host interface function.

PRECAUTIONS

This section provides general precautions for using the Programmable Terminal.

The information contained in this section is important for the safe and reliable application of the Programmable Terminal. You must read this section and understand the information contained before attempting to set up or operate a Programmable Terminal.

1 Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems (an electrical engineer or the equivalent).

- Personnel in charge of introducing FA systems into production facilities.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and connecting FA systems.
- Personnel in charge of managing FA systems and facilities.

2 General Precautions

The user must operate the product according to the performance specifications described in the operation manuals.
Before using the product under conditions which are not described in the manual or applying the product to nuclear control systems, railroad systems, aviation systems, vehicles, combustion systems, medical equipment, amusement machines, safety equipment, and other systems, machines and equipment that may have a serious influence on lives and property if used improperly, consult your OMRON representative.
Make sure that the ratings and performance characteristics of the product are sufficient for the systems, machines, and equipment, and be sure to provide the systems, machines, and equipment with double safety mechanisms.
This manual provides information for using the Programmable Terminal. Be sure to read this manual before attempting to use the software and keep this manual close at hand for reference during operation.

⚠ **WARNING** It is extremely important that Programmable Terminals and related devices be used for the specified purpose and under the specified conditions, especially in applications that can directly or indirectly affect human life. You must consult with your OMRON representative before applying Programmable Terminals to the above-mentioned applications.

⚠ **WARNING** Do not use input functions such as PT keys for applications where danger to human life or serious damage is possible, or for emergency switch applications.

3 Safety Precautions

Read these safety precautions carefully and make sure you understand them before using the Programmable Terminal so that you can use it safely and correctly.

Safety Conventions and their Meanings

This operation manual uses the following conventions and symbols to indicate cautions, warnings, and dangers in order to esure safe use of the NT4S/NT15S/NT18S.
The caustions, warnings, and dangers shown here contain important information related to safety. This instructions in these cautions, warnings, and dangers must be observed.

The conventions used and their meanings are presented below.

⚠ WARNING Indicates information that, if not heeded, could possibly result in loss of life or serious injury.

⚠ CAUTION Indicates information that, if not heeded, could result in relatively serious or minor injury, damage to the product, or faulty operation.

Explanation of Symbols

This manual uses the following symbols to indicate notes and hazardous situations.

 Notes for the User

 General Danger

 Specific Danger

SECTION 1
Siemens PU (AS511)

1.1 General Information

The operating terminals can be conveniently connected to the Siemens PLCs S90 - S155 thus making the operating terminals the perfect man-machine-interface for your Siemens PLC.
The operating terminal is connected to the PU (programming unit) interface of the PLC. An additional communications module is not necessary. The data communication on the interface is handled by the PU protocol AS511.
Connections to the following Simatic S5 PLC types are supported:

SIMATIC S5-90U (8-bit CPU)
SIMATIC S5-95U (8-bit CPU)
SIMATIC S5-100U
- CPU 100 (8-bit CPU)
- CPU 102 (8-bit CPU)
- CPU 103 (8-bit CPU)

SIMATIC S5-115U
- CPU 941 (8-bit CPU)
- CPU 942 (8-bit CPU)
- CPU 943 (8-bit CPU)
- CPU 944 (8-bit CPU)
- CPU 945 (8-bit CPU with 20-bit address range)

SIMATIC S5-135U
- CPU 922 (16-bit CPU)
- CPU 928 (16-bit CPU)
- CPU 928B (16-bit CPU)

SIMATIC S5-155U
- CPU 948 (20-bit CPU)

Connection to the following Simatic S5 PLC type is **not supported**:

SIMATIC S5-135U
- CPU 921 (16-bit CPU)

The software components of the system are fully adapted to the parameters and marginal conditions of the PU interface.

This offers the user the following advantages:

- Random read and write access to all data within the PLC. Data from existing PLC programs can be displayed and modified directly on the operating terminal. Since the communication data do not need to be stored in a specified address area or data type area, it is not necessary to adapt the PLC program to the operating terminal in any respect.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- Use of a PU multiplexer allows simultaneous connection of the operating terminal and Siemens programming unit (PU).

- No configuration required within the PLC.

- The PU protocol is handled entirely by the firmware of the PLC. A PLC program (function blocks, etc.) in the PLC is not required to handle the communication.

- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission. An electrically isolated, noise-immune interface hardware in accordance with the 20 mA current loop interface standard permits the application even in a harsh industrial environment.

- The parameters of the interface SER1 are assigned in the programming software in a protocol-specific manner and are stored in the application description. The parameters can also be modified in the setup mask or any other I/O mask of the terminal.

- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program.

1.2 Technical Description

The PU protocol is used to connect the operating terminal to the Siemens PLCs. The PU protocol AS511 allows random read and write access to all PLC data. Any byte-structured data types can also be accessed in bit-mode. The size of the address area depends on the respective PLC.

Access to the individual bytes of a data word within a data block is also possible.

A read access must be performed before individual bits or bytes of a data word within a data block can be accessed for a write operation. Subsequently, a write access is possible to the entire data structure. When accessing individual bits or bytes, special care needs to be taken to ensure that neither the terminal nor the PLC modify individual bits within one byte (or individual bits within one data word, respectively).

1.3 Protocol Parameters Siemens PU AS511

To ensure proper communication, the parameters must **not** be altered.

Baud rate:	300, 600, 1200, 2400, 4800, **9600**, 19200, 38400, 375000, 500000 Baud
Parity:	none, **even**, odd
Data length:	5, 6, 7, **8**
Stop bits:	1, 1.5, **2**
Handshake:	**no handshake**, Hardware, Software

The operating terminal adapts to the default parameters of the PU interface. It is therefore **not** necessary to modify the interface parameters in the PLC. This allows communication between the operating terminal and Siemens PLC without requiring any configuration.

1.3.1 Parameter Maximum Waiting Time for Response

This timer indicates the length of time that the operating terminal (master) waits for the response from the PLC (slave).
Permitted values are in the range of 0 ms to 65535 ms.
The default value is 500 ms.

This value must be increased when using a CPU with two PU interfaces (see „Guidance for the employment of CPUs with 2 PU interfaces").

1.3.2 Parameter Delay Until Connection Setup

Specifies the period of time that the terminal allows to elapse after an unsuccessful attempt to establish the connection and before making another attempt.
Permitted values are in the range of 5000 ms to 255000 ms.
The default value is 10000 ms.

1.3.3 Parameter Fast Data Block Access

The base address for each data block being used is determined only once and this information is stored temporarily in a local buffer with 10 positions. Any subsequent accesses continue to operate with the information stored in the local buffer. The information in the buffer is erased upon restarting the terminal or upon a resynchronization after a communication error.

Important:
In this case, do not modify the size of data blocks or compress the PLC memory while the connection between the terminal and the PLC is still being established! If a terminal and a programming unit are simultaneously connected to the PLC by means of a multiplexer, then any value of the data block that is altered via the programming unit and transferred into the PLC, also results in a change of the address location of the data block. In this case, the cache should be deactivated.

1.4 Data Type Structure

a) Alphanumerical Texts
Are stored in the memory byte for byte in ascending address order.

b) Counter
A distinction is made between variables which have been assigned a counter address and variables which have been assigned another PLC address.

Counter address
When accessing counter addresses, the count value is interpreted in the binary format and the control bits of the counter are masked out. Therefore, to avoid control bits from being erased, counter addresses should be accessed in the **read-mode** only.

All other addresses
The count value is interpreted in BCD-code. This allows transfer of this value within the PLC program to the counter by means of the accumulator. This service should be used for indirect write-operations of count values since the values are available in the Siemens conforming format.

c) Timer
Timer functions consist of a time value and a time base. The terminal operates with imaginary unsigned 4-byte variables even though the data stored in the PLC comprise only 2 bytes.

When read-accessing timers, the terminal converts the time value and time base into a terminal-internal unsigned 4-byte variable which represents the time value in reference to the time base of 0.01 second.

Example : A range of 10 (time base is 1.0 second) and a time value of 999 are represented or edited in the terminal by the value 99900. Scaling of this value to other value ranges is possible by specifying a factor and divisor within the variable definition.

Before writing a timer variable to the PLC, a terminal-internal unsigned 4-byte value is converted to a time value and the smallest possible time base.
In addition, a distinction is made between variables which have been assigned a timer address or another PLC address.

Timer address
When accessing timer addresses, the time value is interpreted in binary format. To avoid timer control bits from being erased, this service should be used in the **read-mode** only.

All other addresses
The time value is interpreted BCD-coded. Because the values are available in the Siemens conforming format, this service should be used for indirect write-operations of time values.

d) Floating Point Number
The data are interpreted in the Siemens floating point format.

e) Binary Variables with a Length of 1, 2 or 4 Bytes
Data with a length of 2 bytes are interpreted in the PLC-conforming byte order for words.
Data with a length of 4 bytes are interpreted in the PLC-conforming byte order for long words.

1.4.1 Data Types

Direct accessing of the following data types is possible:

E	input bits	(bit access, read-only)
A	output bits	(bit access)
M	flag bits	(bit access)
EB	input bytes	(byte access, read-only)
AB	output bytes	(byte access)
MB	flag bytes	(byte access)
EW	input word	(word access, read-only)
AW	output word	(word access)
MW	flag word	(word access)
ED	input double word	(double word access, read-only)
AD	output double word	(double word access)
MD	flag double word	(double word access)
DW	data word	(word access)
DL	data word, left (high)	(word access)
DR	data word, right (low)	(word access)
DD	data double word	(double word access)
T	timer	(word access, read-only)
Z	counter	(word access, read-only)

The size of each data area depends on the CPU of the PLC.

1.4.2 Special Simatic Data Formats

The following data formats are supported in the editors:

KB 0 to 255
Variable in byte format

KF -32768 to +32767
Variable in 16-bit fixed point number format

KH 0000 to FFFF
Variable in 4-digit hexadecimal number format

DH 00000000 to FFFFFFFF
Variable in 8-digit hexadecimal number format

KC !! to zz (2 ASCII-characters each)
Variable represented by 2 characters in ASCII format

KT 000.0 to 999.3
Variable represented as a time value

KZ 000 to 999
Variable represented as a count value

KG ±1.2*10-38 to .±3.4*10+38
Variable in 32-bit floating point number format

KM 00000000 00000000 to 11111111 11111111
Variable in bit pattern format

1.5 Additional Functions

In addition to the random read and write access to PLC variables, a 11-byte memory area is specified in the application description as a poll area. The location of this memory area can also be specified in the application description.

The only marginal conditions regarding this memory area is that the PLC must be able to access in bit-mode and the terminal in byte-mode and the memory area must be contiguous.

Access modes of the PLC to the poll area are:

Byte-structured Memory Mapping
The maximum size for the data area is 11 bytes.
Example:
The cyclic poll area is set to flag byte (MB) 12 in the programming system.

Access to the PLC occurs via:

Byte address	MB	Description
Byte address +0	MB12	Write coordination byte
Byte address +1	MB13	Message channel low-byte
Byte address +2	MB14	Message channel high-byte
Byte address +3	MB15	Function key LED 1 to 4
Byte address +4	MB16	Function key LED 5 to 8
Byte address +5	MB17	Function key LED 9 to 12
Byte address +6	MB18	Function key LED 13 to 16
Byte address +7	MB19	Function key LED 17 to 20
Byte address +8	MB20	Function key LED 21 to 24
Byte address +9	MB21	Function key LED 25 to 28
Byte address +10	MB22	Function key LED 29 to 32

Word-structured Memory Mapping
The maximum size for the data area is 6 words or 12 bytes.

Example:
The cyclic data area is set to DW21 in the programming system.

Word address	DW	High-byte	Low-byte
Word address +0	DW21	Write coordination byte	Reserved
Word address +1	DW22	Message channel high-byte	Message channel low-byte
Word address +2	DW23	Function key LED 1 to 4	LED 5 to 8
Word address +3	DW24	Function key LED 9 to 12	LED 13 to 16
Word address +4	DW25	Function key LED 17 to 20	LED 21 to 24
Word address +5	DW26	Function key LED 25 to 28	LED 29 to 32

1.6 Error Messages

Code			
	1	E_SLAVE_NOT_READY	Slave not ready
	2	E_PROTOKOL	Sequence of the packets
	3	E_FRAME	Protocol frame error
	4	E_TIMEOUT	Timeout error
	6	E_PARITY	Parity error
	7	E_SEND_ABORT	Send process aborted
	8	E_REC_ABORT	Receive process aborted
	9	E_BUF_SIZE	Insufficient cyclic buffer
	10	E_NO_DEFINE	No cyclic data defined
	12	E_DEFINE	Cyclic data already defined
	15	E_NO_PROTOCOL	Selected protocol is not supported
	16	E_OVERRUN	Receive buffer overrun
	17	E_NAK	NAK from PLC despite repetition
	40	E_SYS_ADDRESS	Undefined system variable

Siemens-specific error messages

	50	E_BST_RANGE	Address outside of the range of the data block
	51	E_RECEIVE_COUNT	Number of data received is incorrect
	52	E_FUN_NOT_KNOWN	Unknown function
	53	E_WRONG_MODE	Wrong mode of operation
	54	E_DATA_BLOCK_NOT_KNOWN	Data block does not exist
	55	E_HIGH_LEVEL	Communication has reached higher level
	56	E_LOW_LEVEL	Communication has reached lower level
	57	E_MESSAGE_CONNECT	Invalid feedback received from the PLC during the connection setup phase
	58	E_MESSAGE_TRANSFER	Invalid feedback received from the PLC during the transfer phase
	59	E_MESSAGE_DISCONNECT	Invalid feedback received from the PLC during the disconnect phase
	60	E_CPU_TYP_921	A CPU of the type 921 has been detected. The operating terminal does not support this CPU type.
	61	E_WRONG_ADDR	Wrong addressing used for job. The requested address level is not fully available.

1.7 Guidance for the Employment of CPUs with 2 PU Interfaces

CPUs with 2 PU interfaces (e.g. 928B with PU module) are used during system commissioning thus keeping the PU interface available for program debugging.

Please note that the interfaces are interacting (see Siemens CPU manual). For example, if a PLC program is analysed by means of the STATUS-function on the first interface of the CPU, this will cause the speed of protocol handling on the second interface to decrease significantly. With the STATUS-function, the PLC program can, so to speak, be operated in the single-step-mode, i.e. the result of every program line is displayed on the screen.

In order to maintain the connection, the timeout period must be increased to 5 seconds. Using the regular timeout period results in the terminal generating a communication error message.

In this case, each transmission requires 3-4 seconds, in contrast to a regular communication cycle which requires approximately 60 ms. This means a slow-down of the interface by a factor of 50!

SECTION 2
GE Fanuc SNP

2.1 General Information

The operating terminals allow for a simple connection to every GE FANUC PLC of the 90 series thus making the operating terminals the perfect man-machine-interface for your FANUC PLC.

The operating terminal is connected to the **Programming and Communications Port** of the PLC-CPU. An additional communications module is not necessary. The data communication on the interface is handled by the FANUC-Series 90 protocol (SNP-protocol).

The software components of the system are fully adapted to the parameters and marginal conditions of the SNP interface.

This offers the user the following advantages:

- Random write and read access to any data within the PLC. Data of existing PLC programs can be displayed and modified directly in the operating terminal. It is not necessary to adapt the PLC program to the operating terminal in any respect since it is not required that communications data be stored in a specified address area or data type area.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- Configuration of a bus topology is possible. Use of a SNP multiplexer allows simultaneous connection of the operating terminal and the hand-held programming unit (HHP) or the Logicmaster (PC with LM90 software), respectively.

- Only a minimum of configuration is required in the PLC. Point-to-point connections require no configuration at all in the PLC. A bus topology will require setting of only one parameter for the communication, namely the slave identifier (CPU-ID) for the PLC. This slave identifier is entered once with the LM90.

- The SNP protocol is handled entirely by the operating system of the PLC. A PLC program in the PLC is not required for the handling of the communication.

- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission. A noise-immune interface hardware in accordance with the RS485 interface standard permits the application even in a harsh industrial environment.

- The parameters of the interface SER1 are assigned in the programming software in a protocol-specific manner and are stored in the application description. Modifying of the parameters is possible in the setup mask or in each other I/O-mask of the terminal at any time.

- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program (e.g. %M3 corresponds to flag 3).

2.2 Protocol Parameters GE Fanuc SNP

The parameters of the interface SER1 are set to the following values:

Baud rate:	300, 600, 1200, 2400, 4800, 9600, **19200**, 38400, 357000, 500000 Baud
Parity:	none, even, **odd**
Data length:	5, 6, 7, **8** bits
Stopbits:	**1**, 1.5, 2
Handshake:	No handshake, Hardware, Software

The default parameters of the programming software are printed **bold**.
The operating terminal adapts to the default parameters of the SNP protocol. Therefore it is generally not necessary to alter the interface parameters in the PLC.

2.2.1 Parameter Maximum Waiting Time for Response

This timer indicates the length of time that the operating terminal (master) will wait for the response from the PLC (slave).
Permitted values are in the range of 50 ms to 65535 ms.
The default value is 1000 ms.

2.2.2 Parameter Delay Until Connection Setup

Specifies the period of time that the terminal allows to elapse after an unsuccessful attempt to establish the communication and before making another attempt.
Permitted values are in the range of 5000 ms to 255000 ms.
The default value is 10000 ms.

2.3 Configuring the PLC

Before any communication between the operating terminal and FANUC-PLC is possible, it will be necessary to set the CPU ID parameter with the aid of a LM90.

2.3.1 CPU - ID

The operating terminal considers the PLC a slave and, thus, references it via a slave number. This slave number is assigned to the variable during the creation of the application definition. Valid slave numbers range from value 1 to value 253.

Having defined a slave number 1 to 253 in the application definition will require that the same CPU-ID consisting of 3 characters must be specified in the PLC. Any blank digits will be filled in with zeros (slave number 23 corresponds to the CPU-identifier 023).

It is also possible to work without a CPU-identifier by specifying the slave identifier 254 in the application definition. In this event it must be ensured that the connection between terminal and PLC is a point-to-point-connection.

2.4 Data Types

User references are bit, byte, or word-structured.

Reference type	Sign	Orientation
Discrete Inputs	%I	Bit / Byte
Discrete Outputs	%Q	Bit / Byte
Temporary Coils	%T	Bit / Byte
Internal Coils	%M	Bit / Byte
System Status References	%SA, %SB, %SC, %S	Bit / Byte
Discrete Globals	%G	Bit / Byte
Analog Inputs	%AI	Word / Double word
Analog Outputs	%AQ	Word / Double word
Registers	%R	Word / Double word

The size of each reference area is governed by the size of the PLC's CPU.

2.5 Additional Functions

In addition to the random write and read access to PLC variables, a memory area comprising 11 or 12 bytes is specified in the application definition as poll area. The location of this memory area is specified in the application definition.
Only marginal conditions regarding this memory area:
- the PLC must be able to access in bit-mode and the SNP in byte-mode
- the memory area must be contiguous.

Byte-structured Memory Mapping
The data area comprises a maximum of 11 bytes
The start address must be located on a physical byte boundary.

The following start addresses are possible in accordance with (8 * n) + 1 ; n = 0, 1 , 2 : M1, M9, M17.

Example: Cyclic Data starts at %M1

MSB	Description	LSB
%M8	Write Coordination Byte	%M1
%M16	Message Channel Low byte	%M9
%M24	Message Channel High byte	%M17
%M32	LED 1 to 4	%M25
%M40	LED 5 to 8	%M33
%M48	LED 8 to 12	%M41
%M56	LED 13 to 16	%M49
%M64	LED 17 to 20	%M57
%M72	LED 21 to 24	%M65
%M80	LED 25 to 28	%M73
%M88	LED 29 to 32	%M81

Word-structured Memory Mapping
The data area comprises a maximum of 6 words or 12 bytes.

Example: The cyclic data on %R1

Register	High byte	Low byte
%R1	Write Coordination Byte	Reserved
%R2	Message Channel High byte	Message Channel Low byte
%R3	LED 1 to 4	LED 5 to 8
%R4	LED 9 to 12	LED 13 to 16
%R5	LED 17 to 20	LED 21 to 24
%R6	LED 25 to 28	LED 29 to 32

2.6 Error Messages

Code	1	E_SLAVE_NOT_READY	Slave not ready or incorrect CPU-ID
	2	E_PROTOKOL	Sequence of the packets
	3	E_FRAME	Character frame error
	4	E_TIMEOUT	Timeout error
	5	E_CRC_BCC	CRC error
	6	E_PARITY	Parity error
	7	E_SEND_ABORT	Abort send process
	8	E_REC_ABORT	Abort receive process
	9	E_BUF_SIZE	Insufficient cyclic buffer
	10	E_NO_DEFINE	No cyclic data defined
	12	E_DEFINE	Cyclic data already defined
	14	E_SLAVE_ADDRESS	Invalid slave address (e.g. 0x00 master-address)
	15	E_NO_PROTOCOL	Selected protocol is not supported
	16	E_OVERRUN	Receive buffer overrun
	18	E_NAK_0	BCC or parity error from PLC
	19	E_NAK_0	Overrun or framing error from PLC
	20	E_NAK_0	Sequence error from PLC
	21	E_NAK_0	Bad Next Message Length error from PLC
	40	E_SYS_ADDRESS	Undefined system variable or invalid slave number 255

Fanuc-specific error messages

Code	50 - 138	Major error status code	The error number consists of the constant 50 and the error status
	178 - 25	Minor error status code of the SNP-partner-module	

SECTION 3
Mitsubishi FX

3.1 General Information

The operating terminals allow for a simple connection to every Mitsubishi PLC of the FX series thus making the operating terminals the perfect man-machine-interface for your Mitsubishi PLC.

The operating terminal is connected to the **programming unit interface** of the PLC. An additional communications module is not necessary. The data communication on the interface is handled by the Mitsubishi FX protocol.

The software components of the system are fully adapted to the parameters and marginal conditions of the interface.

This offers the user the following advantages:

- Random write and read access to any data within the PLC. Data of existing PLC programs can be displayed and modified directly in the operating terminal. It is not necessary to adapt the PLC program to the operating terminal in any respect since it is not required that communications data be stored in a specified address area or data type area.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- No configuration is required in the PLC.

- The protocol is handled entirely by the operating system of the PLC. A PLC program in the PLC is not required for the handling of the communication.

- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission. A noise-immune interface hardware in accordance with the RS485 interface standard permits the application even in a harsh industrial environment.

- The parameters of the interface SER1 are assigned in the programming software in a protocol specific manner and are stored in the application description. Modifying of the parameters is possible in the setup mask or in each other I/O-mask of the terminal at any time.

- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program (e.g. M3 corresponds to flag 3).

3.2 Technical Description

The interfacing of the operating terminal to the Mitsubishi-FX Series- PLCs is effected by means of the FX protocol.

The FX protocol allows random read and write access to all PLC (programmable logic controller) data. Any byte-structured data types can also be accessed in bit-mode. The size of the address area depends on the respective PLC.

3.3 Protocol parameters Mitsubishi FX

The parameters of the interface SER1 are set to the following values:

Baud rate:	300, 600, 1200, 2400, 4800, **9600**, 19200, 38400, 375000, 500000 Baud
Parity:	none, **even**, odd
Data length:	5, 6, **7**, 8 bits
Stopbits:	**1**, 1.5, 2
Handshake:	**No Handshake**, Hardware, Software

The default parameters of the programming software are printed in **bold**.

3.3.1 Parameter Maximum Waiting Time for Response

This timer indicates the length of time that the operating terminal (master) will wait for the response from the PLC (slave).
Permitted values are in the range of 0 ms to 65535 ms.
The default value is 5000 ms.

3.3.2 Parameter Delay Until Connection Setup

Specifies the period of time that the terminal allows to elapse after an unsuccessful attempt to establish the communication and before making another attempt.
Permitted values are in the range of 5000 ms to 255000 ms.
The default value is 10000 ms.

3.4 Data Types

Type	Description	Access
S	Step flag operand (status)	access in bit and byte mode
X	Inputs	access in bit and byte mode
Y	Outputs	access in bit and byte mode
M	Flags	access in bit and byte mode
T	Timer flags	access in bit and byte mode
T	Time Value	access in word mode
C	Count Flags	access in bit and byte mode
C	Count Values	access in word (double word) mode
M	Flag (also Special Flag)	access in bit and byte mode
D	Data Register (also Special Data Register)	access in word (double word) mode

3.5 Additional Functions

In addition to the random write and read access to PLC variables, a memory area comprising 11 or 12 bytes is specified in the application description as cyclic data area. The location of this memory area is specified in the mask definition and must be located on a byte boundary.

Only marginal conditions regarding this memory area:
- the PLC must be able to access in bit-mode
- the memory area must be contiguous.

Byte-structured Memory Mapping

The start address must be located on a physical byte boundary.

The following start addresses are possible in accordance with (8 * n) n = 0, 1, 2 : M0, M8 M16.

Example: Cyclic Data on M0

MSB	Description	LSB
M7	Write coordination byte	M0
M15	Message channel low byte	M8
M23	Message channel high byte	M16
M31	LED 1 to 4	M24
M39	LED 5 to 8	M32
M47	LED 9 to 12	M40
M55	LED 13 to 16	M48
M63	LED 17 to 20	M56
M71	LED 21 to 24	M64
M79	LED 25 to 28	M72
M87	LED 29 to 32	M80

Word-structured Memory Mapping

Example: Cyclic data on D1

Register	High-byte	Low-byte
D1	Write coordination byte	Reserved
D2	Message channel high-byte	message channel low-byte
D3	LED 1 to 4	LED 5 to 8
D4	LED 9 to 12	LED 13 to 16
D5	LED 17 to 20	LED 21 to 24
D6	LED 25 to 28	LED 29 to 32

3.6 Error Messages

Code			
	1	E_SLAVE_NOT_READY	Slave not ready or incorrect CPU-ID
	2	E_PROTOKOL	Sequence of the packets
	3	E_FRAME	Character frame error
	4	E_TIMEOUT	Timeout error
	5	E_CRC_BCC	CRC error
	6	E_PARITY	Parity error
	7	E_SEND_ABORT	Abort send process
	8	E_REC_ABORT	Abort receive process
	9	E_BUF_SIZE	Insufficient cyclic buffer
	10	E_NO_DEFINE	No cyclic data defined
	12	E_DEFINE	Cyclic data already defined
	15	E_NO_PROTOCOL	Selected protocol is not supported
	16	E_OVERRUN	Receive buffer overrun
	17	E_NAK	NAK from the PLC
	40	E_SYS_ADDRESS	Undefined system variable

SECTION 4
SINEC L1

The operating terminals allow for a simple connection to the Siemens S95 - S155 programmable controllers thus making the operating terminals the perfect man-machine-interface for your Siemens PLC.

With the programmable controllers 95U, 100U (CPU 103) and 155U, the operating terminal can be connected to the PU interface of the PLC. An additional communications module is not necessary.
With the programmable controllers 115U, 135U and 155U, a connection is possible via a CP530 communications processor.
In either case, the operating terminal is the master of the bus.

The software components of the system are fully adapted to the parameters and marginal conditions of the SINEC L1 protocol.
This offers the user the following advantages:

- Random write and read access to any data within the PLC. Data of existing PLC programs can be displayed and modified directly in the operating terminal. It is not necessary to adapt the PLC program to the operating terminal in any respect, since it is not required that communications data be stored in a specified address area or data type area.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- Use of a PU multiplexer allows simultaneous connection of the operating terminal and a hand-held programming unit (PU).

- Only a minimum of configuration is required for the installation of the supplied function blocks into the PLC.

- Minimal increase of the cycle time of the PLC.

- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission. A noise-immune interface hardware in accordance with the 20 mA current loop interface standard permits the application even in a harsh industrial environment.

- The parameters of the interface SER1 are assigned in the programming software in a protocol-specific manner and are stored in the application description. Modifying of the parameters is possible in the setup mask or in each other I/O-mask of the terminal at any time.

- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program.

4.1 Protocol Parameters SINEC L1

To ensure proper communication, the parameters must **not** be altered.

Baud rate:	**9600** Baud
Parity:	**even**
Data length:	**8**
Stopbits:	**1**
Handshake:	**no handshake**

The operating terminal adapts to the default parameters of the SINEC L1 interface. Therefore it is **not** necessary to modify the interface parameters in the PLC.

4.2 Data Types

Random read and write access is possible to any data in the PLC. All byte-structured data types can also be accessed in bit-mode. The size of the address area depends on the PLC being used.

Access to the individual bytes of a data word within a data block is also possible.

Direct access is possible to the following data types:

E	input bits	(read-only)
A	output bits	
M	flag bits	
EB	input bytes	(read-only)
AB	output bytes	
MB	flag bytes	
EW	input word	(read-only)
AW	output word	
MW	flag word	
ED	input double word	(read-only)
AD	output double word	
MD	flag double word	
DW	data word	
DL	data word, left-hand (high)	
DR	data word, right-hand (low)	
DD	data double word	
T	timer	(read-only)
Z	counter	(read-only)

The size of each data area is governed by the CPU of the PLC.

4.2.1 Data Type Structure

a) Alphanumerical Text
Is stored in the memory byte for byte in ascending address order.

b) Counter
A distinction is made between variables which have been assigned a counter address and variables which have been assigned another PLC address.

Counter address
When accessing counter addresses, the count value is interpreted in the binary format, the control bits of the counter are masked out. Therefore, to avoid control bits from being erased, counter addresses should be accessed in the **read-mode** only.

All other addresses
The count value is interpreted in BCD-code. This allows the transfer of this value within the PLC program to the counter by means of the accumulator. This function should be used for indirect write-operations of count values since the values are available in the Siemens conformal format.

c) Timer
Timer functions consist of a time value and a time base. The terminal operates with imaginary unsigned 4-byte variables, even though the data stored in the PLC comprise only 2 bytes.

When read-accessing the timer, the terminal converts the time value and time base into a terminal-internal unsigned 4-byte number, which represents the time value in reference to the time base of 0.01 seconds.

Example: A range of 10 (time base is 1.0 second) and a time value of 999, are represented or edited, respectively, in the terminal by the value 99900. Scaling of this value to other value ranges is possible by specifying a factor and divisor within the variable definition.

Before writing a timer variable to the PLC, the time value and the smallest possible time base are formed from the terminal-internal unsigned 4-byte value.

In addition, a distinction is made between variables which have been assigned a timer address and variables which have been assigned another PLC address.

Timer address
When accessing timer addresses, the time value is interpreted in the binary format. To avoid timer control bits from being erased, this access should occur in the **read-mode** only.

All other addresses
The time value is interpreted BCD-coded. This access should be used for indirect write-operations of time values since the values are available in the Siemens conformal format.

d) Floating Point Number
The data are interpreted in the Siemens floating point format.

e) Binary Variables with a Length of 1, 2 or 4 Bytes
Data with a length of 2 bytes are interpreted in the PLC-conformal byte order for words.
Data with a length of 4 bytes are interpreted in the PLC-conformal byte order for long words.

4.3 Configuring the PLC

Before a communication between the operating terminal and the Siemens PLC is possible, the supplied function blocks must be installed and executed during the PLC start-up or at cyclic intervals, respectively, and a communications data block must be specified in the RAM of the PLC.

4.3.1 Status Data Area

In addition to the random read and write access to PLC variables, an area comprising 6 words (DW70 - DW75) is available in the data block which allows the PLC to influence the terminal (status data area). The transmission of the entire area to the terminal is activated as soon as the value 128 is written into DL69. After the transmission, DL69 will be reset to 0 by the FB201.

The status data area is word-structured and has been assigned to fixed addresses:

DL70 :	Write coordination byte
DR70 :	Reserved
DW71 :	Sequential message channel
DW72 - DW76 :	LED-bits for function keys

4.4 Error Messages

Code 1 E_SLAVE_NOT_READY .....The slave address has been sent, however, no response has been received within the specified time period of 5 seconds.

Possible causes for the error:
- Incorrect slave address.
- No hardware connection.

2 E_FRAME .....The SIO has detected a framing error.

Possible cause for the error:
- Incorrect interface parameter in the terminal or in the PLC.

4 E_TIMEOUT.....After successful completion of the addressing phase and after sending the telegram from the terminal to the PLC, the response telegram from the PLC has not been received within the bus monitoring time period of 300 ms.

5 E_CRC_BCC.....Checking of BCC1 or BCC2 has revealed a transmission error.

6 E_PARITY.....Detection of a parity error.

16 E_OVERRUN.....The interrupt service routine has not picked up the received characters quickly enough from the SIO.

Possible cause for the error:
- Programming error in the terminal.

17 E_NAK .....Error bit in ZBS has been set.

Possible causes for the error:
- The PLC has detected that an error (e.g. parity error) has occurred during transmission of the telegram from the Terminal to the PLC.
- PLC is in the stop-mode.

40 E_SYS_ADDRESS .....Undefined system variable or invalid slave number 255

51 E_POLL_NR .....After transmitting a write or read request to the function block in the PLC, no response has been received within the maximum number of poll attempts (100).

Possible causes for the error:
- Function block FB201 in the PLC is not being executed.
- PLC is in the stop mode.
- PLC is overloaded and is therefore not capable of assembling the response quickly enough (unlikely).

52 E_RECV_ADDR.....The address received during the addressing phase does not correspond to the transmitted address.

Possible cause for the error:
- Transmission error not identified despite of parity check.

53 E_QV_BIT.....The request bit (ZBS) for inter-node communication is set to logic 1.

Possible cause for the error:
- The SINEC L1 interface in the PLC is not accessed by the function block FB201 exclusively (PLC-application program error).

54 E_RECV_LEN.....The length of the data received, exceeded 64 bytes.

55 E_DEST.....The destination in the telegram from the PLC to the terminal does not correspond to 0 (master address).

Possible cause for the error:
- The SINEC L1 interface in the PLC is not accessed by the function block FB201 exclusively (PLC-application program error).

56 E_DATA_TYPE .....An attempt was made to read a data type which is not supported by the function block FB201.

Possible causes for the error:
- Function block FB201 has been reduced in order to save memory space and no longer supports all data types.
- Attempt to write-access inputs, timers or counters.
- Mask definition in the FLASH-EPROM contains errors.

4.5 SINEC L1-Data Handling Software for S5 115U

4.5.1 Installation

The interfacing of the operating terminals to the Simatic S5 115U via the PU-interface by means of the SINEC L1 protocol requires the installation of the supplied data handling software into the PLC. To install the software, carry out the following steps:

- The four supplied function blocks FB200, FB201, FB202 and FB203 must be installed.

- A data block (L1DB) must be created in the RAM of the PLC. The minimum size required for this data block is 77 data words. The data block can either be created with the aid of the programming unit or through the start-up OBs. This process must be carried out before one of the function blocks mentioned above is executed for the first time.

- FB200 must be executed through the start-up blocks OB21 and OB22. FB200 will parameterize the system data area for the SINEC L1 communication and perform the required initialization of data in L1DB.

- FB201 must be executed through OB1. FB201 controls the data exchange with the operating terminal. When necessary, it will execute FBs 202 or 203 for this process. These two function blocks may not be executed directly by the application software.

The function blocks use flag words FW252 and FW254 as scratch flags. This means that if you wish to retain the information in the flags, it will be necessary to transfer the contents of these flag words before the communications function blocks are executed.

4.5.2 Interface to the Application Software

The interface to the PLC application software is not of a very complex design. This is due to the fact that the supplied function blocks handle the entire data exchange process with the operating terminal via the send mail box and receive mail box. The function blocks merely need to be executed as described above.

The function blocks also control every access, read-access as well as write-access, to data in the PLC. Again, there is no need for an application software for this process.

The only task remaining for the application software is the transfer of the status data area. For this process, an area comprising 6 words (DW70 to DW75) has been reserved in the communications data block L1DB. This area allows the PLC to influence the terminal.
The entire area will be transferred to the terminal whenever the value 1 is written into DL69. After the transmission has been completed, the supplied function blocks will reset DL69 to 0. Thus DL69 represents a coordination byte which is used to control the points of time at which the status data area is to be transferred or indicates whether the transfer has already been effected.

The coordination byte can also be used by the application software for the function communication monitoring. This is possible since the coordination byte is reset to 0 after a specified period of time only if the PLC is still connected to the operating terminal. The timeout period should logically be greater than 1 second.

All other data of the L1DB, with the exception of the data mentioned above, may not be accessed by the application software.

4.5.3 Initialization Function Block FB200

The process of executing FB200 through the start-up blocks OB21 and OB22 must be carried out only once. Prior to this, however, the communications data block L1DB must have been created since access to certain words of this data block is required for the execution process.

This function block initializes the system data area of the PLC using SINEC L1 communication parameters.

The following values are entered for the parameters:

PU-number:	not relevant here
Slave-number:	as specified during programming of the terminal
KBE:	DW1 in L1DB
KBS:	DW0 in L1DB
Receive mail box:	DW36 to DW68 in L1DB
Send mail box:	DW2 to DW34 in L1DB

In addition, FB200 performs the required initialization processes in L1DB.

The only parameter to be defined for FB200 is the number of the communications data block L1DB.

4.5.4 Communications Function Block FB201

FB201 must be executed by OB1 at cyclic intervals. The handling of the data exchange with the operating terminal is performed entirely by this function block. This process involves that the FB, each time it is being executed, verifies whether a read or write request has been received from the terminal and carries out such a request, if necessary. To carry out a read or write request, it will execute FB202 or FB203, respectively, which are responsible for reading or writing to variables, respectively.

In the event that the terminal has not sent a request, the FB will check whether the status data area is to be transferred to the terminal and will, if necessary, carry out this process.

The only parameter to be defined for FB201 is the number of the communications data block L1DB.

4.6 SINEC L1 Data Handling Software for S5 135U with CP530

4.6.1 Installation

The interfacing of the operating terminals to the Simatic S5 135U via the CPU 530 requires the installation of the supplied data handling software into the PLC. To install the software, carry out the following steps:

- The four supplied function blocks FB200, FB201, FB202 and FB203 must be installed.

- A data block (L1DB) must be created in the RAM of the PLC. The minimum size required for this data block is 93 data words. The data block can either be created with the aid of the programming unit or through the start-up OBs. This process must be carried out before one of the function blocks mentioned above is executed for the first time. The block must be assigned the number 15, since this is the number that is being used for the parameterization of the SEND and RECEIVE data handling block FB201. In the event that DB-number 15 has already been used elsewhere, another option is to alter the DBNR parameter when executing the data handling blocks in FB 201.

- FB200 must be executed through start-up blocks OB20, OB21 and OB22. FB 200 will handle the synchronization with the CP530 and also carry out the required initialization of the data in L1DB.

- FB201 must be executed through OB1. FB 201 controls the data exchange with the operating terminal. When necessary, it will execute FBs 202 or 203 for this process. These two function blocks may not be executed directly by the application software.

The function blocks use flag words FW248 to FW254 as scratch flags. This means that if you wish to retain this information, it is necessary to transfer the contents of these flag words before the communications function blocks are executed.

4.6.2 Interface to the Application Software

The interface to the PLC application software is not of a very complex design which is due to the fact that the supplied function blocks handle the entire data exchange process with the operating terminal via the send mail box and receive mail box. The function blocks merely need to be executed as described above.

The function blocks also control every access, read as well as write access, to data in the PLC. Again, there is no need for an applications software for this process.

The only task remaining for the application software is the transfer of the status data area. For this process, an area comprising 6 words (DW70 to DW75) has been reserved in the communications data block L1DB. This area allows the PLC to influence the terminal.
The entire area will be transferred to the terminal whenever the value 80H is written into DL69. After the transmission has been completed, the supplied function blocks will reset DL69 to 0. Thus, DL69 represents a coordination byte which is used to control the points of time at which the status data area is to be transferred or indicates whether the transfer has already been completed.

The coordination byte can also be used by the application software for the function communication monitoring. This is possible since the coordination byte is reset to 0 after a specified period of time only if the PLC is still connected to the operating terminal. The timeout period should logically be greater than 1 second.

All other data of the L1DB, with the exception of the data mentioned above, may not be accessed by the application software.

4.6.3 Initialization Function Block FB200

The process of executing the FB200 through the start-up blocks OB20, OB21 and OB22 must be carried out only once. Prior to this, however, the communications data block L1DB must have been created since access to certain words of this data block is required for the execution process.

This function block synchronizes the CP530. The interface number which is required for the synchronization must be assigned to the FB as parameter.

FB200 also carries out the initialization processes required in L1DB. The number (SSNR) of the L1DB must be assigned to the FB as parameter as well.

The error output (FEHL) is set, whenever an error has occurred during the process of synchronization.

4.6.4 Communications Function Block FB201

FB201 must be executed by OB1 at cyclic intervals. The handling of the data exchange with the operating terminal is performed entirely by this function block. This process involves that the FB, each time it is being executed, verifies whether a read or write request has been received from the terminal and carries out such a request if necessary. To carry out a read or write request, FB will execute FB202 or FB203, respectively, which are responsible for reading or writing to variables, respectively.

In the event that the terminal has not sent a request, the FB will verify whether the status data area is to be transferred to the terminal and will, if necessary, carry out this process.

The only parameter to be defined for FB201 is the number of the communications data block L1DB.

4.6.5 Parameterizing the CP530

Addressing of the communications processor CP530 must occur using the interface number specified during the execution of FB200.

In addition, the CP530 must be operated as a SINEC L1 slave. The slave number assigned to the communications processor must coincide with the number defined during the programming of the operating terminal (preferably 1).

Programming of polling lists and interrupt lists is not required for a SINEC L1 slave.

SECTION 5
3964/RK512

5.1 General Information

Via the serial procedure 3964, the operating terminals can be connected to a programmable controller.

The logical part of the 3964 protocol, RK512, is adapted to the communication with a Siemens PLC via a CP525, or compatible, communications processor.

The software components of the system are fully adapted to the parameters and marginal conditions of the protocol 3964/ RK512.

This offers the user the following advantages:

- Random read and write access to any data within the PLC. Data of existing PLC programs can be displayed and modified directly in the operating terminal. It is not necessary to adapt the PLC program to the operating terminal in any respect since it is not required that communications data be stored in a specified address area or data type area.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- Simultaneous connection of the operating terminal and the programming unit (PU) is possible.

- Only a minimum of configuration is required for the installation of the data handling block required in the PLC in addition to the protocol function blocks.

- Minimal increase of the cycle time in the PLC.

- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission. A noise-immune interface hardware in accordance with the 20 mA current loop interface standard permits the application even in a harsh industrial environment.

- The parameters of the interface SER1 are assigned in the programming software in a protocol specific manner and are stored in the application description. Modifying of the parameters is possible in the setup-mask or in each other I/O-mask of the terminal at any time.

- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program.

5.2 Technical Description

The terminal is always the active partner which either requests data from or sends data to the partner.
Direct read-access is possible to all PLC data.
Direct write-access is limited to data blocks only.

The installation of the supplied function block allows an indirect write-access to all PLC data types.

All byte-structured data types can also be accessed in bit-mode.
It is also possible to access all individual bytes of a data word within a data block.

5.3 Protocol parameters 3964/RK512

The parameters of the interface SER1 are set to the following values:

Baud rate:	300, 600, 1200, 2400, 4800, **9600**, 19200, 38400, 375000, 500000 Baud
Parity:	none, **even**, odd
Data length:	5, 6, 7, **8** Bits
Stop bits:	1, 1.5, **2**
Handshake:	**no handshake**, hardware, software

The default parameters of the programming system are printed in **bold**.

The following protocol parameter settings are also available for interface **SER1:**
- communications block number
- data word offset
- coordination flag number
- coordination flag bit
- protocol with/without coordination flag
- block check
- CPU number

The interface parameters of the communications module must comply with the parameters of the operating terminal.

5.3.1 PLC Configuration

To allow write-access to all PLC data, it is merely necessary to install the supplied function block and to execute it at cyclic intervals.
In addition, a coordination flag must be defined and a data block must be created as communications block with a size of 128 bytes.
This communications block and coordination flag must be specified to the supplied function block as parameter.

5.4 Data Type Structure

a) Alphanumerical Texts
Are stored in the memory byte for byte in ascending address order.

b) Counter
A distinction is made between variables which have been assigned a counter address and variables which have been assigned another PLC address

Counter address
When accessing counter addresses, the count value is interpreted in the binary format, the control bits of the counter are masked out. Therefore, to avoid control bits from being erased, counter addresses should be accessed in the **read-mode** only.

All other addresses
The count value is interpreted in BCD-code. This allows the transfer of this value within the PLC program to the counter by means of the accumulator. This function should be used for indirect write-operations of count values since the values are available in the Siemens conformal format.

c) Timer
Timer functions consist of a time value and a time base. The terminal operates with imaginary unsigned 4-byte variables, even though the data stored in the PLC comprise only 2 bytes.

When read-accessing the timer, the terminal converts the time value and time base into a terminal-internal unsigned 4-byte number, which represents the time value in reference to the time base of 0.01 second.

Example: A range of 10 (time base is 1.0 second) and a time value of 999, are represented or edited, respectively, in the terminal by the value 99900. Scaling of this value to other value ranges is possible by specifying a factor and divisor within the variable definition.

Before writing a timer variable to the PLC, the time value and the smallest possible time base are formed from the terminal-internal unsigned 4-byte value.

In addition, a distinction is made between variables which have been assigned a timer address or another PLC address.

Timer address
When accessing timer addresses, the time value is interpreted binary format. To avoid timer control bits from being erased, this access should occur in the **read-mode** only.

All other addresses
The time value is interpreted BCD-coded. This access should be used for indirect write-operations of time values since the values are available in the Siemens conformal format.

d) Floating Point Number
The data are interpreted in the Siemens floating point format.

e) Binary Variables with a Length of 1, 2 or 4 Bytes
Data with a length of 2 bytes are interpreted in the PLC-conformal byte order for words.
Data with a length of 4 bytes are interpreted in the PLC-conformal byte order for long words.

5.4.1 Data Types

Direct accessing of the following data types is possible:

E	input bits	(bit access)
A	output bits	(bit access)
M	flag bits	(bit access)
EB	input bytes	(byte access)
AB	output bytes	(byte access)
MB	flag bytes	(byte access)
EW	input word	(word access)
AW	output word	(word access)
MW	flag word	(word access)
ED	input double word	(double word access)
AD	output double word	(double word access)
MD	flag double word	(double word access)
DW	data word	(word access)
DL	data word, left-hand (high)	(word access)
DR	data word, right-hand (low)	(word access)
DD	data double word	(double word access)
T	timer	(word access)
Z	counter	(word access)

The size of each data area is governed by the CPU of the PLC.

5.4.2 Special Simatic-Data Formats

The following data formats are supported in the editors:

KB 0 to 255
Variable in byte-format

KF -32768 to +32767
Variable in 16-bit fixed point number-format

KH 0000 to FFFF
Variable in 4-digit hexadecimal number-format

DH 00000000 to FFFFFFFF
Variable in 8-digit hexadecimal number-format

KC !! to zz (2 ASCII-characters each)
Variable represented by 2 characters in ASCII-format

KT 000.0 to 999.3
Variable represented as time value

KZ 000 to 999
Variable represented as count value

KG ±1.2*10-38 to ±3.4*10+38
Variable in 32-bit floating point number-format

KM 00000000 00000000 to 11111111 11111111
Variable in bit pattern-format

5.5 Additional Functions

In addition to the random write and read access to PLC variables, a memory area comprising 11 (12) bytes is specified in the mask definition as poll area. The location of this memory area can also be specified in the mask definition.

Only marginal conditions regarding this memory area:
- the PLC must be able to access in bit-mode and the terminal in byte-mode
- the memory area must be contiguous.

Byte-structured Memory Mapping

The data area comprises a maximum of 11 bytes

Example: The cyclic poll area is set to flag byte MB12 in the programming system.

Access to the PLC occurs via:

Byte address	MB	Description
Byte address +0	MB12	Write Coordination Byte
Byte address +1	MB13	Message Channel Low byte
Byte address +2	MB14	Message Channel High byte
Byte address +3	MB15	Function Key LED 1 to 4
Byte address +4	MB16	Function Key LED 5 to 8
Byte address +5	MB17	Function Key LED 9 to 12
Byte address +6	MB18	Function Key LED 13 to 16
Byte address +7	MB19	Function Key LED 17 to 20
Byte address +8	MB20	Function Key LED 21 to 24
Byte address +9	MB21	Function Key LED 25 to 28
Byte address +10	MB22	Function Key LED 29 to 32

Word-structured Memory Mapping

The data area comprises a maximum of 6 words or 12 bytes.

Example: The cyclic data area is set to DW21 in the programming system.

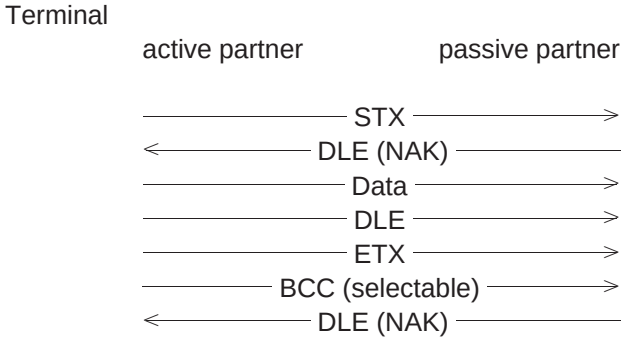
Word address	DW	High byte	Low byte
Word address +0	DW21	Write Coordination Byte	Reserved
Word address +1	DW22	Message Channel High	Message Channel Low
Word address +2	DW23	Function Key LED 1 to 4	LED 5 to 8
Word address +3	DW24	Function Key LED 9 to 12	LED 13 to 16
Word address +4	DW25	Function Key LED 17 to 20	LED 21 to 24
Word address +5	DW26	Function Key LED 25 to 28	LED 29 to 32

5.6 3964 Procedure

To initiate the communication setup, the active partner will transmit the signal STX (02h) upon which the partner will have to respond with DLE (10h) within the specified acknowledgement delay period (2 s).
Subsequently, the procedure data are transmitted, the receipt of which is monitored by a character delay time (220 ms).
After the data have been transmitted, the passive partner will acknowledge the receipt with DLE.

In the event of errors, the passive partner will transmit a NAK (15h).
In the event of errors during the communication setup, the active partner will make up to 3 attempts to establish the communication.

If the passive partner transmits a NAK after receipt of the data, the active partner will make up to 6 attempts to establish the communication and to transmit the data.



Please note:
If the character 10_{16} is to be transmitted but not to be evaluated as a DLE, the transmitting device will add another 10_{16}.
If the receiving device detects a 10_{16} (DLE) twice, it will accept 10_{16} only once and will not evaluate it as a DLE control character.

5.6.1 Block Check BCC

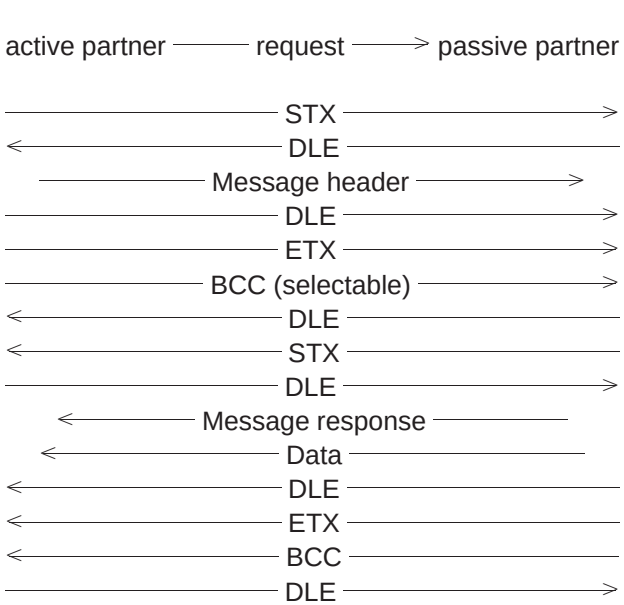
With the block check method (3964R), a block check character is created and added to the end of the block.

The BCC is formed through a logic XOR of all characters with the exception of the start character STX (02_{16}).

5.6.2 Logical Part of the Procedure 3964, RK512

A logical part, which complies with the Siemens protocol RK512 to a great extent, has priority over the physical part of the procedure.
This part governs the type and contents of the data part.

5.7 Message Request of Data



5.7.1 Structure Message Header (10 bytes) Request of Data

	Function	ASCII	Hex	Comment
1. Byte	Message		00	always 00
2. Byte	identifier		00	always 00
3. Byte	Data direction E		45	E = Request
4. Byte	Command			see below
5. Byte	2 Byte			see below
6. Byte	Source			
7. Byte	2 Byte			see below
8. Byte	Number of bytes			
9. Byte	Coordination flag number FF indicates no			
10. Byte	coordination flag bit			FF Coord. flag

5.7.2 Data Specification in the Message Header

The data types of the PLC addressed by the terminal are implemented in the message header by the Bytes 4 - 8 as illustrated below.

Source	Access	Data type	Command Byte 4	Word-Parameter Byte 5+6	Byte-Parameter Byte 5	Byte 6	Number in Byte 7+8
Input	Bit	E	E		Offset	Bit-No	1 Byte
Output	Bit	A	A		Offset	Bit-No	1 Byte
Flag	Bit	M	M		Offset	Bit-No	1 Byte
Input	Byte	EB	E	Offset			n Bytes
Output	Byte	AB	A	Offset			n Bytes
Flag	Byte	MB	M	Offset			n Bytes
Input	Word	EW	E	Offset			n Bytes
Output	Word	AW	A	Offset			n Bytes
Flag	Byte	MB	M	Offset			n Bytes
Input	D-Word	ED	E	Offset			n Bytes
Output	D-Word	AD	A	Offset			n Bytes
Flag	D-Word	MD	M	Offset			n Bytes
D-Block	Word	DW	D		DB	DW	n Words
High-Byte DB	Byte	DL	D		DB	DW	1 Word
Low-Byte DB	Byte	DR	D		DB	DW	1 Word
D-Block	D-Word	DD	D		DB	DW	n Words
Timer	Word	T	T	Offset			n Words
Counter	Word	Z	Z	Offset			n Words

5.7.3 Coordination Flag

A flag bit is specified in the message header which is used by the receiving device to monitor the receipt of data.
The monitoring function is deactivated if the value for the coordination flag is FF_{16}, FF_{16}.

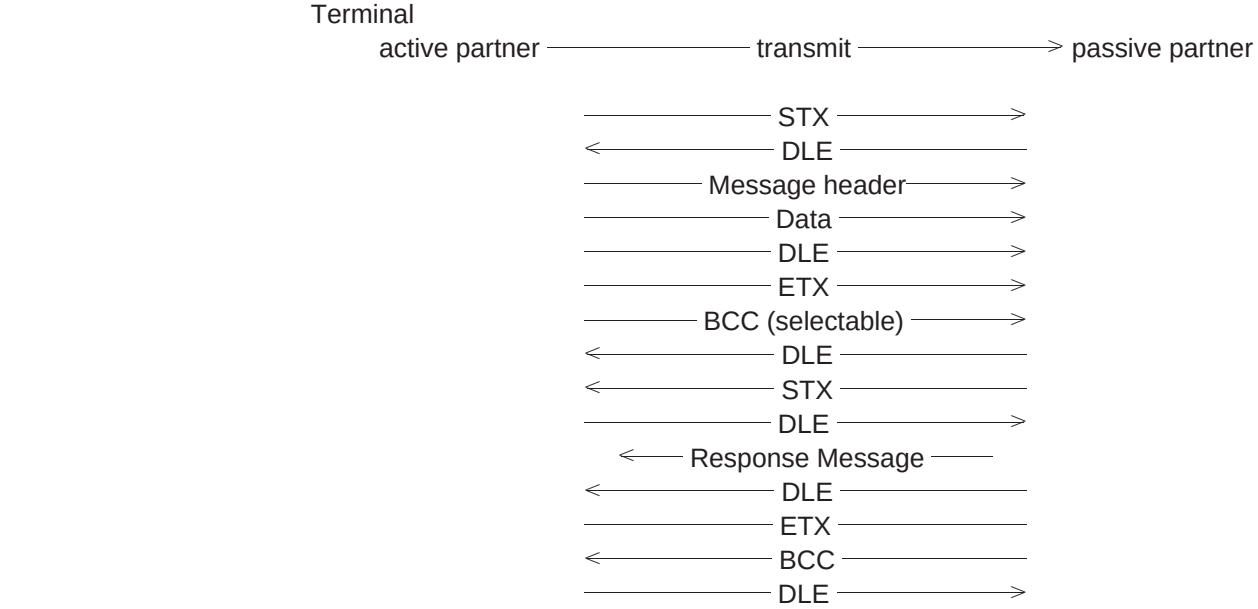
If a coordination flag is specified, it will be set in the passive partner upon receipt of data.
Once this flag has been set, processing of the data received will be initiated.
After the data have been processed, the flag will be reset again.
If this flag is still set upon the receipt of data, the passive partner will transmit a response message to the active partner thereby indicating the respective error.

NOTE:
For connections to a Siemens communications module CP524/525, the coordination flag in DB1 must be entered into the list of interprocessor communication flags.

5.7.4 Structure 4-Byte Sized Response Message

	Function	ASCII	Hex	Comment
1. Byte	Message		00	always 00
2. Byte	identifier		00	always 00
3. Byte				always 00
4. Byte	Error code		xx	

5.8 Message Transmission of Data



5.8.1 Structure Message Header (10 bytes) Transmission of Data

	Function	ASCII	Hex	Comment
1. Byte	Message		00	always 00
2. Byte	identifier		00	always 00
3. Byte	Data direction	A	41	A = Transmission
4. Byte	Command	D	44	D = Data block
5. Byte	2 Byte Destination			Data block
6. Byte				Data word
7. Byte	2 Byte number of data			Number of words
8. Byte				
9. Byte	Coordination flag number			FF indicates without
10. Byte	Coordination flag bit			FF Coord. flag

The destination for data to be transmitted is always a data block.

5.8.2 Special Features of the Protocol 3964R

A write-access to data is possible in data blocks only.
I.e. only the data type DW allows a direct write-access to the respective destination.

To allow a write-access to all data yet, the data are transmitted to a defined data block, the communications block.

In this case, the destination, comprising 4 bytes, must be added in front of the data prior to transmission.

5.8.3 Assignment of Bytes 1-4

Destination	Access	Data type Byte 1	Command Byte 2	Data-Number Byte 3	Byte-Parameter Byte 4		Word-Parameter Bytes 3+4
Input	Bit	I	10	1 Byte	Bit-No.	Offset	
Output	Bit	Q	11	1 Byte	Bit-No.	Offset	
Flag	Bit	F	12	1 Byte	Bit-No.	Offset	
Input	Byte	IB	1	n Bytes		Offset	
Output	Byte	QB	2	n Bytes		Offset	
Flag	Byte	FB	3	n Bytes		Offset	
Input	Word	IW	1	n Bytes		Offset	
Output	Word	QW	2	n Bytes			Offset
Flag	Word	FW	3	n Bytes			Offset
Input	Word	ID	1	n Bytes			Offset
Output	Word	QD	2	n Bytes			Offset
Flag	Word	FD	3	n Bytes			Offset
HIGH-Byte DB	Byte	DL	26	1 Byte	DB	DW	
LOW-Byte DB	Byte	DR	27	1 Byte	DB	DW	
Counter	Word	C	20	n Bytes			Offset
Timer	Word	T	21	n Bytes			Offset

A write-access to these data types is principally carried out with a coordination flag. Just like the communications block, the coordination flag must be defined such that the settings in the terminal and passive device are in agreement.

Part of a program installed in the receiving device, denoted in the PLC as data handling block, will monitor the coordination flag.

If the coordination flag is set, the data will be processed by the data handling block in accordance with the first 4 bytes in the communications block.
After the data have been processed, the coordination flag will be erased thereby allowing further transmissions.

5.9 Protocol 3964R - Restrictions

The protocol 3964 allows data comprising a maximum of 128 bytes to be transmitted per message. No further messages are transmitted or processed.

5.10 Function Block for Siemens 115 U

The supplied function block FB186 can be implemented in a Siemens 115 U.

The function block processes data from the terminal which are to be transmitted to the PLC.

The first 4 bytes of the data which are written to the defined communications data block are interpreted in accordance with the table ”Assignment of Bytes 1-4”.

The function block FB186 supports the following commands from the table ”Assignments of Bytes 1-4”:

2	Output Byte
3	Flag Byte
11	Output Bit
12	Flag Bit.

When activating the function block, the communications data block and the coordination flag must be specified as parameter. These parameters must comply with the settings in the terminal.

The defined communications data block must comprise a size of 128 bytes or 64 data words, respectively.

5.11 Application Example for CP525 in 115U

In this example:
The communications data block is DB33 starting at DW0.
The coordination flag is M100.3.
Flag 50 is to be written to.

The coordination flag, also referred to as interprocessor communication flag by Siemens, must be activated on the CP525 via a hardware jumper (see reference material (CP525).

In addition, the interprocessor communication flag must be defined in DB1 of the 115U. An illustration is shown below:

```
 DB1 :
DW 0:   KH=4D41 \
     1: KH=534B >        ”MASK01”     Header identifier
     2. KH=3130 /

DW 3:   KH=CA00                      Output-interprocessor com-
                                     munication flag
     4: KF=100                       MB100

DW 5:   KH=EEEE                      End identifier
```

The following structure must be implemented in OB1. This block is executed at cyclic intervals.

```
      :     .
      :     .
      :SPA FB245      Execution of the RECEIVE-block
NAME:REC-ALL

ANZW:MW220            The interprocessor communication flag number is
                      entered in the indication word parameter of the
           .          RECEIVE-block
           .
      :L   MB221      If the defined interprocessor communication flag is in
                      the LOW-Byte
      :L   KF+100
      .I=F
      :S   M 100.3    it must be set here.
      :  .
      :  .
      :SPA FB186      Evaluation block
                      Processes the data of the communications data block
                      and resets flag 100.3.
```

5.12 Initialization of Module K43 of EBERLE PLS514

For communication with the module K43 of an EBERLE PLS 514 the following parameters must be set:

For TTY-Interfacing:

Baud rate:	9600
Parity:	even
Data bits:	8
Stop bits:	1
Handshake:	no handshake

Coordination flag:	none
Spec. PLC-communication:	none
DB:	0
DW:	0
Block check:	Yes
CPU-number within the PLC:	0
Floating point numbers in IEEE-Format:	no

For RS232-Interfacing:

Baud rate:	9600
Parity:	even
Data bits:	8
Stop bits:	1
Handshake:	hardware

Coordination flag:	none
Spec. PLC-communication:	none
DB:	0
DW:	0
Block check:	Yes
CPU-number within the PLC:	0
Floating point numbers in IEEE-Format:	no

For timer, BCD numbers and counter always double words (32 bit) must be used, e.g. DW0.

The initialization of the module K43 is done with the values containing in the file INI.IL. The values containing in this file must be set as follows.

```
{Initialisierung des K43 Moduls}

*********************************************************}

{       Programmbaustein:       INI.PBS}

{       Versionsnummer:         01 01}

{       Keine Änderung im Programm}

{  *********************************************************}

{ }
{  *********************************************************}
{       Initialisierung des K43}
{  *********************************************************}

{       Dieses Programm dient zur Initialisierung des }
{       Kommunikationsmoduls K43.}
{       Die Initialisierung erfolgt selbständig nach-}
{       dem Einschalten der Versorgungsspannung der }
{       PLS 514.}

{       Im ersten Zyklus der PLS 514 werden die benötigten}
{       Merker initialisiert und das Initialisierungsprogramm }
{       aufgerufen.}

{       Im Initialisierungsprogramm werden die Parameter für}
{       das Modul K 43 übergeben. Dieses Programm ist nicht }
{       notwendig, wenn die Default-Einstellung verwendet wird.}

{       Nach der Initialisierung wird automatisch in die }
{       Betriebsart Datenaustausch mit K 43 umgeschaltet.}

{       Maßgebend für die jeweilige Betriebsart des K 43 ist das}
```

```
{       Kommandodigit. }
{       Das Kommandodigit ist das Digit 3 der Ebene 0.}

{       Die einzelnen Bits haben folgende Bedeutung.}

{   Bit   3   2   1   0}
{       ___________________}
{       !   !   !   !   !}
{       ! X ! X ! X ! X !}
{       !   !   !   !   !}
{       ‾‾‾‾‾‾‾‾‾}
{         !   !   !   !}
{         !   }
{         !   !   !   0  = Kein SW-Reset }
{         !   1   !   1  = SW-Reset}
{         !   }
{         !   !   X      = Keine Bedeutung}
{         !}
{         !   0          = K 43 Betriebsart passiv}
{         !   1          = K 43 sendet über V.24 TTY}

{         0              = Initialisierung}
{         1              = Datenaustausch mit K 43 aktiv}

{       Im ersten Zyklus rücksetzen des Initialisierungs-}
{       schieberegisters und der benötigten Merker }
{       Um den Selbsttest des Schnittstellenmoduls K 43 }
{       zu überbrücken ist eine Einschaltverzögerung in }
{       der PLS 514 notwendig. Diese wird über den Hilfs-}
{       zähler #Z0 bis #Z2 erzeugt.}
{       ‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾}

L       %ZK1
LD      %K 0
=D      INIREG
=D      Z0
=D      Z1
=D      Z2
=D      DW_ZV_0
=D      DW_ZV_1
=D      DW_ZV_2
=D      DW_ZV_3

L       %MS10
ZV      Z0
ZV      Z1
ZV      Z2

L       %K 1
LD      Z2
GL      %K 2
LD      INIREG
GL      %K 0
S       INIREG0

{       Initialisierung des Protokolls:}
{       ==============================}

{       Für die Initialisierung des Protokolls haben die }
{       Digits 0 .. 2 der Ebene 0 und die Digits der Ebene 1}
{       die jeweils beschriebene Bedeutung}


DAL     K43_E0
```

```
L       INIREG0
A       DACK
=       %NOP

{       Definition des Kommandodigits Digit 3 der Ebene 0}
{       ————————————————————————————}

{       Einstellen der Datenübertragung zum K 43 auf }
{       Initialisierung-Modus}

{       Digit 0xx3}

{   Bit   3   2   1   0}
{       ———————————————————}
{       !   !   !   !   !}
{       ! X ! X ! X ! X !}
{       !   !   !   !   !}
{       —————————}
{         !   !   !   !}
{         !   }
{         !   !   !   0  = Kein SW-Reset }
{         !   1   !   1  = SW-Reset}
{         !   }
{         !   0          = K 43 Betriebsart passiv}
{         !   1          = K 43 sendet über V.24 TTY}

{         0              = Initialisierung}
{         1              = Datenaustausch mit K 43 aktiv}

{       Eingabe der Konstanten 00 in das Digit 0xx3 }
{       d.h. Initialisierung des K 43 }
{               ohne SW-Reset}
{               Protokoll initialisieren gemäß Ebene 0, }
{               Digit 0...2 und Ebene 1, Digit 0...7}

LD      %K 0
=D      KOMMAND

{       Initialisierung über Ebene 0 Digit 0,1 und 2:}
{       ——————————————————————————}
{       Digit 0xx0}

{   Bit   3   2   1   0}
{       ———————————————————}
{       !   !   !   !   !}
{       ! X ! X ! X ! X !}
{       !   !   !   !   !}
{       —————————}
{         !   !   !   !}
{         !   }
{         !   0   0   0  =   110 Baud }
{         !   0   0   1  =   300 Baud }
{         !   0   1   0  =   600 Baud }
{         !   0   1   1  =  1200 Baud }
{         !   1   0   0  =  2400 Baud }
{         !   1   0   1  =  4800 Baud }
{         !   1   1   0  =  9600 Baud }
{         !   1   1   1  = 19200 Baud       }

{         0              = Datenformat 7 Bit }
{         1              = Datenformat 8 Bit }

{       Eingabe der Konstanten 14 in das Digit 0xx0 }
```

```
{       d.h. 9600 Baud, Datenformat 8 Bit}

LD      %K 14
=D      Data00


{       Digit 0xx1}
{       ─────}
{ }
{   Bit   3   2   1   0}
{       ___________________}
{       !   !   !   !   !}
{       ! X ! X ! X ! X !}
{       !   !   !   !   !}
{       ─────────}
{         !   !   !   !}
{         !   !   !}
{         !   !   !   0  = kein Parity-Bit, keine Prüfung  (none)}
{         !   !   !   1  = Parität wie Bit 1.1 und 1.2}
{         !   !   }
{         !   0   0   1  = Prüfung auf ungerade Parität   (odd)}
{         !   1   0   1  = Prüfung auf gerade Parität     (even)}
{         !   }
{         !   0   1   1  = Parity-Bit immer auf „1“ gesetzt (mark)}
{         !                 keine Prüfung                 }
{         !   1   1   1  = Parity-Bit immer a. „0“ gesetzt (space)}
{         !                 keine Prüfung                 }

{         0             =  1 Stop Bit}
{         1             =  2 Stop Bit}

{       Eingabe der Konstanten 05 in das Digit 0xx1 }
{       d.h. gerade Paritöt, 1 Stop Bit }
{ }
LD      %K 5
=D      Data01


{       DIGIT 0xx2}
{       ─────}

{   Bit   3   2   1   0}
{       ___________________}
{       !   !   !   !   !}
{       ! X ! X ! X ! X !}
{       !   !   !   !   !}
{       ─────────}
{         !   !   !   !}
{         !   !   !}
{         !   !   !   0  = RTS/CTS eingeschaltet}
{         !   !   !   1  = RTS/CTS ausgeschaltet}
{         !   !   }
{         !   !   0      = V 24}
{         !   !   1      = TTY}
{         !   }
{         !   0          = aktives Senden mit niedriger Priorität}
{         !   1          = aktives Senden mit hoher Priorität}

{         0             = Protokoll 3964 aktiv}
{         1             = Protokoll 3964 R aktiv}

{       Eingabe der Konstanten 09 in das Digit 0xx2 bei RS232 }
{       Eingabe der Konstanten 11 in das Digit 0xx2 bei TTY }
{       d.h. Handshake XON/XOFF, V24-Schnittstelle, }
```

```
{       niedrige Priorität, Protokoll 3964 R aktiv.}

LD      %K 11
=D      Data02

{       Initialisierung der Ebene 1}
{       ==========================}

{       Die Initialisierung der Datenbausteine erfolgt im }
{       hexadezimalen Code. Bei der Angabe des entsprechenden}
{       Zeichens wird jede Stelle einzeln eingegeben. }
{       D.h. zwei Halbbytes ergeben ein Zeichen und müssen demzu-}
{       folge in zwei Digits eingetragen werden.}
{  }
{       Initialisierung des 0. Datenbausteine}
{       ————————————————————}

{       Digit 1xx0  niederwertiges Halbbyte des 0. Datenbausteins}
{       Digit 1xx1  höherwertiges  Halbbyte des 0. Datenbausteins}
{           }
{  —>  z.B. 1. Datenbaustein auf 32 dez. => 20 hex. }
{ }
LD      %K 0
=D      Data10
LD      %K 2
=D      Data11

{       Initialisierung des 1. Datenbausteins}
{       ————————————————————}
{                                                 }
{       Digit 1xx2  niederwertiges Halbbyte des 1. Datenbausteins}
{       Digit 1xx3  höherwertiges  Halbbyte des 1. Datenbausteins}

{  —>  z.B.  1. Datenbausteins auf 33 dez. => 21 hex. }

LD      %K 1
=D      Data12
LD      %K 2
=D      Data13

{       Initialisierung des 2. Datenbausteins}
{       ————————————————————}

{       Digit 1xx4  niederw. Halbbyte des 2. Datenbausteins}
{       Digit 1xx5  höherwert. Halbbyte des 2. Datenbausteins}

{  —>  z.B. 2. Datenbausteins auf 34 dez. => 22 hex. }
{   }
LD      %K 2
=D      Data14
LD      %K 2
=D      Data15
{           }
{       Initialisierung des 3. Datenbausteins }
{       ————————————————————}

{       Digit 1xx6  niederw. Halbbyte 3. Datenbausteins }
{       Digit 1xx7  höherwert. Halbbyte des 3. Datenbausteins }
{           }
{  —>  z.B. 3. Datenbausteins  auf  35 dez. => 23 hex. }

LD      %K 3
=D      Data16
LD      %K 2
```

```
=D      Data17

{       Beenden der Initialisierung}
{       ———————————————}

SL      INIREG
DAS     K43_E0
DAS     K43_E1
DAL     K43_E0
R       %NOP

{       Start der Datenübertragung}
{       ==========================}

{       Nachdem die Initialisierung beendet ist, wird die}
{       Datenübertragung durch setzen des Kommandodigits }
{       freigegeben.}

{       Bedeutung des Kommandodigits wahrend der Datenübertragung}

{   Bit   3   2   1   0}
{         ———————————————————}
{         !   !   !   !   !}
{         ! X ! X ! X ! X !}
{         !   !   !   !   !}
{         —————————}
{           !   !   !   !}
{           !   }
{           !   !   !   0  = Kein SW-Reset }
{           !   1   !   1  = SW-Reset}
{           !   }
{           !   0          = K 43 Betriebsart passiv}
{           !   1          = K 43 sendet über V.24 TTY}

{           0              = Initialisierung}
{           1              = Datenaustausch mit K 43 aktiv}


{       Definition des Kommandodigits Digit 3 der Ebene 0}
{       durch die Konstante 08}
{       d.h. Datenübertragung mit Blockübertragungs-Mode}
{       ————————————————————————————}

L       INIREG1
A       DACK
=       %NOP
LD      %K 8

=D      KOMMAND

{       Weiterschalten der Initialisierung}
{       ——————————————————}

SL      INIREG
DAS     K43_E0
DAS     K43_E1
DAL     K43_E0
R       %NOP

{  ********************************************************}
{       Reset des K 43}
{  ********************************************************}

{       SW_Reset des K 43 über den Merker #Bed_Res}
```

```
{       durch setzen des Kommandodigits }

{       Bedeutung des Kommandodigits}

{   Bit   3   2   1   0}
{       ___________________}
{       !   !   !   !   !}
{       ! X ! X ! X ! X !}
{       !   !   !   !   !}
{       _________}
{         !   !   !   !}
{         !   }
{         !   !   !   0  = Kein SW-Reset }
{         !   1   !   1  = SW-Reset}
{         !   }
{         !   0          = K 43 Betriebsart passiv}
{         !   1          = K 43 sendet über V.24 TTY}

{         0              = Initialisierung}
{         1              = Datenaustausch mit K 43 aktiv}

{       Definition des Kommandodigits Digit 3 der Ebene 0}
{       durch die Konstante 9}
{       d.h. Datenübertragung und SW-Reset }
{       ____________________________}

L       Bed_Res
A       Inireg2
A       DACK
=       %NOP

LD      %K 12
=D      KOMMAND

{       Initialisierungsprogramm neu aufrufen}
{       ____________________}

LD      %K 0
=D      INIREG
DAS     K43_E0
DAS     K43_E1
DAL     K43_E0
R       %NOP


Var
Bed_Res %       0007.0  { Bedingung zum Reset K 43}
INIREG  %       0270    { Ini. SCHIEBEREGISTER}
INIREG0 %       0270.0  { Ini. PROTOKOLLS}
INIREG1 %       0270.1  { Ini. DATEN?BERTRAGUNG}
DW_ZV_2 %       0273    { Datenwort Zöhler 2}
DW_ZV_3 %       0274    { Datenwort Zöhler 3}
Z0      %       0275    { Hilfszähler 0}
Z1      %       0276    { Hilfszähler 1}
Z2      %       0302    { Hilfszähler 2}
End_Var
```

5.13 Error Messages

These error messages are displayed on the terminal.

Code			
	1	E_SLAVE_NOT_READY	Slave not ready, no connection
	2	E_PROTOKOL	Invalid character, no repetition
	3	E_FRAME	Byte frame error, despite repetition
	4	E_TIMEOUT	Timeout error
	5	E_CRC_BCC	CRC error, despite repetition
	6	E_PARITY	Parity error, despite repetition
	7	E_SEND_ABORT	Abort send process
	8	E_REC_ABORT	Abort receive process
	9	E_BUF_SIZE	Insufficient cyclic buffer
	10	E_NO_DEFINE	No cyclic data defined
	11	E_DEFINE	Cyclic data already defined
	15	E_NO_PROTOCOL	Protocol is not supported
	16	E_OVERRUN	Overrun
	40	E_SYS_ADDRESS	Undefined system variable
	50	E_QUITTUNG_OPEN	Invalid acknowledge during communication setup
	51	E_QUITTUNG_DATA	Invalid acknowledge after transmission of data
	52	E_NO_RESPONSE	No response message
	53	E_RECEIVE_COUNT	Incorrect number of data received

Errors which are transmitted by the programming controller (PU) via the response message.

61	E_NO_AG	
	10 from PU	No connection to PU
62	E_WRONG_ORDER	
	16 from PU	Invalid command in message header
63	E_INV_DEST	
	20 from PU	Invalid destination has been addressed
64	E_KOO_MERKER_SET	
	50 from PU	Coordination flag is still set
65	E_SEND_COUNT	
	52 from PU	Number of data transmitted does not comply with the specification in the message header
66	E_SYNCH	
	54 from PU	Awaiting response message
70	E_AG	The subcode contains the error which is transmitted in the response message by the PLC

Subcode

10 No connection to PU
12 Start address to high
Using coordination flag as data type not allowed. CPU-number to high.
16 Invalid Opcode
20 DB not available. DB to short
50 Coordination flag still set
52 More or less data received than expected
54 Synchronization error (following telegram expected)

SECTION 6
Klöckner Moeller SUCOM1

6.1 General Information

The operating terminals allow for a simple connection to the Klöckner Moeller PLCs PS 306 and PS 316.

The operating terminal is connected to the PU interface of the PLC. The data communication on the interface is handled by the SUCOM1 protocol.

The software components of the system are fully adapted to the parameters and marginal conditions of the SUCOM1 protocol.

This offers the user the following advantages:

- Random write and read access to any data within the PLC. Data of existing PLC programs can be displayed and modified directly in the operating terminal. It is not necessary to adapt the PLC program to the operating terminal in any respect. It is not required that communications data be stored in a specified address area or data type area.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- No additional configuration required within the PLC.

- Minimal increase of the cycle time of the PLC.

- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission.

- A noise-immune interface hardware in accordance with the RS485 interface standard permits the application even in a harsh industrial environment.

- The parameters of the interface SER1 are assigned in the programming software in a protocol-specific manner and are stored in the application description. Modifying of the parameters is also possible in the setup mask or each other I/O mask of the terminal at any time.

- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program.

6.2 Technical Description

The interfacing of the operating terminal to the Klöckner Moeller PLCs is effected by means of the SUCOM1 programming protocol.

The SUCOM1 protocol allows random read and write access to all PLC data. All data types can also be accessed in bit-mode. The size of the address area depends on the respective PLC.

A read access to the addressed word must occur, before individual bits can be accessed for a write operation. During these accesses, care must be taken to ensure that neither the terminal nor the PLC modify individual bits of the same word.

6.3 Protocol Parameters SUCOM1

The parameters of the interface SER1 are set to the following values:

Baud rate:	300, 600, 1200, 2400, 4800, **9600**, 19200, 38400, 375000, 500000 Baud
Data bits:	5, 6, 7, **8**
Stop bits:	1, 1.5, **2**
Parity:	**none**, even, odd
Handshake:	**no handshake**, hardware, software

The default values of the programming system are printed in **bold**.

6.4 Data Types

I	input bits	(bit access)
Q	output bits	(bit access)
M	flag bits	(bit access)
IB	input bytes	(byte access)
QB	output bytes	(byte access)
MB	flag bytes	(byte access)
IW	input word	(word access)
QW	output word	(word access)
MW	flag word	(word access)
PS	process status word	(word access)
DS	diagnostic word	(word access)
DZ	diagnostic counter	(word access)

6.5 Additional Functions

In addition to the random write and read access to PLC variables, a memory area comprising 12 bytes is specified in the application description as cyclic poll area.

Only marginal conditions regarding this memory area that the PLC must be able to access in bit-mode and the memory area must be contiguous. The start address of this memory area must be specified in the mask definition and must be located on a word boundary.

The variant ”cyclic data area byte-structured” is not available with this protocol.

Word-structured Memory Mapping

The data area comprises a maximum of 6 words or 12 bytes.

Cyclic data area on MW1

MW	High-byte	Low-byte
MW1	Write coordination byte	Reserved
MW2	Message channel high-byte	Message channel low-byte
MW3	LED 1 to 4	LED 5 to 8
MW4	LED 9 to 12	LED 13 to 16
MW5	LED 17 to 20	LED 21 to 24
MW6	LED 25 to 28	LED 29 to 32

6.6 Error Messages

Code	1	E_SLAVE_NOT_READY	Slave not ready or incorrect CPU-ID
	2	E_PROTOKOL	Sequence of the packets
	3	E_FRAME	Character frame error
	4	E_TIMEOUT	Timeout error
	5	E_CRC_BCC	CRC error
	6	E_PARITY	Parity error
	7	E_SEND_ABORT	Abort send process
	8	E_REC_ABORT	Abort receive process
	9	E_BUF_SIZE	Insufficient cyclic buffer
	10	E_NO_DEFINE	No cyclic data defined
	12	E_DEFINE	Cyclic data already defined
	15	E_NO_PROTOCOL	Selected protocol is not supported
	16	E_OVERRUN	Receive buffer overrun
	17	E_NAK	NAK from the PLC
	40	E_SYS_ADDRESS	Undefined system variable

SUCOM1 Specific Error Messages

	50	E_QUITTUNG_COMM	No acknowledge signal during the communications setup
	51	E_QUITTUNG_DATA	No acknowledge signal has been received after sending an information block
	52	E_NO_DATA	No data upon request

Errors in the Data Section

	53	E_RECEIVE_COUNT	Number of bytes received does not correspond to the number of bytes requested

Possible errors:

- A variable with odd number of bytes is read from a word or double word address within the mask, were the error appears.

SECTION 7
Bosch PU Interfacing via BUEP19

7.1 General Information

The operating terminals allow for a simple connection to the Bosch PLCs. The operating terminal is connected to the respective PLC-module.

The data communication on the interface is handled by the PU protocol BUEP19. Communication is possible with any system implementing the BUEP19 protocol. ZE300 / ZE301 / R300 / R301 / R600.

The software components of the system are fully adapted to the parameters and marginal conditions of the PU interface.

This offers the user the following advantages:

- Random write and read access to any data within the PLC. Data of existing PLC programs can be displayed and modified directly in the operating terminal. It is not necessary to adapt the PLC program to the operating terminal in any respect since it is not required that communications data be stored in a specified address area or data type area.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- No configuration required within the PLC.

- The PU protocol is handled entirely by the firmware of the PLC. A PLC program (function blocks, etc.) in the PLC is not required for the handling of the communication.

- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission. An electrically isolated, noise-immune interface hardware in accordance with the 20 mA current loop interface standard permits the application even in a harsh industrial environment.

- The parameters of the interface SER1 are assigned in the programming software in a protocol-specific manner and are stored in the application description. Modifying of the parameters is also possible in the setup mask or each other I/O mask of the terminal at any time.

- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program.

7.2 Technical Description

The interfacing of the operating terminal to the Bosch PLCs is effected by means of the BUEP19 PU-protocol.

The PU protocol BUEP19 allows random read and write access to all PLC data. Any byte-structured data types can also be accessed in bit-mode. The size of the address area depends on the respective PLC.

A read access must occur, before individual bits or bytes of a flag word can be accessed for a write operation. Subsequently, a write access is possible to the entire data structure. When accessing individual bits or bytes, special care must be taken to ensure that neither the terminal nor the PLC modify individual bits within one byte (or individual bits within one word, respectively).

7.3 Protocol Parameters BUEP19

The operating terminal adapts to the default parameters of the PU interface. It is therefore **not** necessary to modify the interface parameters in the PLC.
To ensure a proper communication, the parameters must **not** be modified.

Baud rate:	**9600** Baud
Parity:	**even**
Data bits:	**8**
Stop bits:	**1**
Handshake:	**no handshake**

7.3.1 Parameter Target Module

To ensure an error-free data transmission, the terminal must be informed of which module is to be connected to interface SER1.
ZE300 / ZE301
R300 / R301
R600

7.3.2 Parameter Block Check

As a default, the protocol BUEP19 implements the data block check method CRC16.
The programming device uses a data block check method in accordance with LRC8.
To avoid difficulties during the development phase during which the PU and the terminal are alternately connected to the PLC, the terminal allows selection of the block check method LRC8.

7.3.3 Parameter Coordination Flag

The protocol permits defining of a coordination flag.

7.4 Data Type Structure

a) Alphanumerical Text
Is stored in the memory byte for byte in ascending address order.

b) Counter
The count value is interpreted in binary format. The maximum value is 8191.

c) Timer
Timer functions consist of a time value and a time base. The terminal operates with imaginary unsigned 4-byte variables, even though the data stored in the PLC comprise only 2 bytes.

When read-accessing the timer, the terminal converts the time value and time base into a terminal-internal unsigned 4-byte number, which represents the time value in reference to the time base of 0.01 second.
Example: A range of 10 (time base is 1.0 second) and a time value of 999, are represented or edited, respectively, in the terminal by the value 99900. Scaling of this value to other value ranges is possible by specifying a factor and divisor within the variable definition.
Before writing a timer variable to the PLC, the time value and the smallest possible time base are formed from the terminal-internal unsigned 4-byte value.

d) Floating Point Numbers
The data are interpreted in the Siemens floating point format.

e) Binary Variables with a Length of 1, 2 or 4 Bytes
Data with a length of 2 bytes are interpreted in the PLC-conformal byte order for words.
Data with a length of 4 bytes are interpreted in the PLC-conformal byte order for long words.

7.4.1 Data Types

Direct accessing of the following data types is possible:

E	Input		B, BY, W, DW
A	Output		B, BY, W, DW
M	Flag		B, BY, W, DW
T	Timer	(R300 and R600 read-only)	W
Z	Counter	(R300 and R600 read-only)	W
D	Data word	(0 to 510)	W, DW
DP	Data buffer	(0 to 510 ; only ZE300)	W

The size of each data area is governed by the CPU of the PLC.

7.5 Additional Functions

In addition to the random write and read access to PLC variables, a memory area comprising 12 bytes is specified in the application description as poll area. The location of this memory area is specified in the application description.

Only marginal conditions regarding this memory area are that the PLC must be able to access in bit-mode and the terminal in **word-mode** and the memory area must be contiguous.

The addresses M, D or DP can be accessed in word-mode.

The data area comprises a maximum of 6 words or 12 bytes.

Example: The cyclic data area is set to DW21 in the programming system.

Word address	**DW**	**High-byte**	**Low-byte**
Word address +0	DW21	Write coordination byte	Reserved
Word address +1	DW22	Message channel high-byte	Message channel low-byte
Word address +2	DW23	Function key LED 1 to 4	LED 5 to 8
Word address +3	DW24	Function key LED 9 to 12	LED 13 to 16
Word address +4	DW25	Function key LED 17 to 20	LED 21 to 24
Word address +5	DW26	Function key LED 25 to 28	LED 29 to 32

7.6 Error Messages

Code	1	E_SLAVE_NOT_READY	Slave not ready
	2	E_PROTOKOL	Sequence of the packets
	3	E_FRAME	Character frame error
	4	E_TIMEOUT	Timeout error
	5	E_CRC_BCC	CRC error
	6	E_PARITY	Parity error
	7	E_SEND_ABORT	Abort send process
	8	E_REC_ABORT	Abort receive process
	9	E_BUF_SIZE	Insufficient cyclic buffer
	10	E_NO_DEFINE	No cyclic data defined
	12	E_DEFINE	Cyclic data already defined
	15	E_NO_PROTOCOL	Selected protocol is not supported
	16	E_OVERRUN	Receive buffer overrun
	40	E_SYS_ADDRESS	Undefined system variable

Bosch-specific error messages

50 E_QUITTUNG_START .....No communication setup
51 E_QUITTUNG_OPEN .....Incorrect acknowledge signal during communication setup
52 E_QUITTUNG_DATA.....Incorrect acknowledge signal to transmitted information block
53 E_NO_RESPONSE_WRONG_ CHAR .....No response message
54 E_TIMEOUT_NO_RESPONSE ....Timeout - no response message
55 E_TIMEOUT_BLOCKZEIT.....Timeout - block time has been exceeded
56 E_TIMEOUT_QUIT_RESPONSE ..Timeout - no acknowledge signal
57 E_ABBRUCH_SPS.....EOT -PLC abort
58 E_RECEIVE_COUNT.....Number of received data is incorrect

PLC-Error

62 E_WRONG_ORDER
32 from the PLC .....write access to T, C, to module not permitted
67 E_WRONG_PARAMETER
37 from the PLC .....incorrect parameter
68 E_CHAR_COUNT
38 from the PLC .....number of bytes received is incorrect accordance with the message header
69 E_SYSTEM
39 from the PLC .....incorrect P1 in the system message
71 E_DIRECTION
41 from the PLC .....direction not defined
72 E_DB_SHORT
42 from the PLC .....DB too small
74 E_DB_NOT_PROG
44 from the PLC .....DB not programmed
76 E_DB_NOT_DEF
46 from the PLC .....DB not defined
78 E_WRONG_TYP
48 from the PLC .....block type unknown
79 E_P2_NULL
49 from the PLC .....parameter 2 is 0
94 E_TELE_TYP
64 from the PLC .....message type incorrect

SECTION 8
Bosch PU Interfacing via BUEP19E

8.1 General Information

The operating terminals allow for a simple connection to the Bosch.
The operating terminal is connected to the respective PLC-module.
The data communication on the interface is handled by the PU protocol BUEP19E.
Communication is possible with any system implementing the BUEP19E protocol.
CL200 / CL350 / CL400 / CL500

The software components of the system are fully adapted to the parameters and marginal conditions of the interface.

This offers the user the following advantages:

- Random write and read access to any data within the PLC. Data of existing PLC programs can be displayed and modified directly in the operating terminal. It is not necessary to adapt the PLC program to the operating terminal in any respect since it is not required that communications data be stored in a specified address area or data type area.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- No configuration required within the PLC.

- The PU protocol is handled entirely by the firmware of the PLC. A PLC program (function blocks, etc.) in the PLC is not required for the handling of the communication.

- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission. An electrically isolated, noise-immune interface hardware in accordance with the 20 mA current loop interface standard permits the application even in a harsh industrial environment.

- The parameters of the interface SER1 are assigned in the programming software in a protocol-specific manner and are stored in the application description. Modifying of the parameters is also possible in the setup mask or each other I/O mask of the terminal at any time.

- The programming system provides a maximum of support to the user in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program.

8.2 Technical Description

The interfacing of the operating terminal to the Bosch PLCs is effected by means of the BUEP19E protocol.

The protocol BUEP19E allows random read and write access to all PLC data. The size of the address area depends on the respective PLC.

8.3 Protocol Parameters BUEP19E

The operating terminal adapts to the default parameters of the interface. It is therefore **not** necessary to modify the interface parameters in the PLC.

Baud rate:	**9600** Baud
Parity:	**even**
Data bits:	**8**
Stop bits:	**1**
Handshake:	**no handshake**

8.3.1 Parameter Target Module

As a target module you can choose between CL500, CL400 or CL200.

8.3.2 Parameter Block Check

As a default, the protocol BUEP19E implements the data block check method CRC16.
The programming device uses a data block check method in accordance with LRC8.
To avoid difficulties during the development phase during which the PU and the terminal are alternately connected to the PLC, the terminal allows selection of the block check method LRC8.

8.3.3 Parameter Coordination Flag

The protocol permits defining of a field coordination flag and a sequence coordination flag.

8.4 Data Type Structure

a) Alphanumerical Text
Is stored in the memory byte for byte in ascending address order.

b) Counter
The count value is interpreted in binary format. The maximum value is 8191.

c) Timer
Timer functions consist of a time value and a time base. The terminal operates with imaginary unsigned 4-byte variables, even though the data stored in the PLC comprise only 2 bytes.

When read-accessing the timer, the terminal converts the time value and time base into a terminal-internal unsigned 4-byte number, which represents the time value in reference to the time base of 0.01 second.
Example: A range of 10 (time base is 1.0 second) and a time value of 999, are represented or edited, respectively, in the terminal by the value 99900. Scaling of this value to other value ranges is possible by specifying a factor and divisor within the variable definition.

Before writing a timer variable to the PLC, the time value and the smallest possible time base are formed from the terminal-internal unsigned 4-byte value.

d) Floating Point Number
The data with a length of 4 bytes are interpreted in the IEEE floating point format.

e) Binary Variables with a Length of 1, 2 or 4 Bytes
Data with a length of 2 bytes are interpreted in the PLC-conformal byte order for words.
Data with a length of 4 bytes are interpreted in the PLC-conformal byte order for long words.

8.4.1 Data Types

Direct accessing of the following data types is possible:
The data types listed below can be accessed in bit, byte or word-mode. The access modes are distinguished by the abbreviations BY, B and W.

Data type				**Access**
E	Input	(0 bis 63)	Byte address	B, BY, W, DW
A	Output	(0 bis 63)	Byte address	B, BY, W, DW
M	Flag	(0 bis 255)	Byte address	B, BY, W, DW
T	Timer	(0 bis 127)	Timer number	BY, W, DW
Z	Counter	(0 bis 127)	Counter number	BY, W, DW
BZ	Status of PLC	RUN/STOP	CL500/400	BY
BZ	Status of PLC	RUN/STOP	CL200	W
DP	Data buffer	(0 bis 511)	Byte address	BY, W, DW
DF	Data field	(0 bis 24575)	Byte address	BY, W, DW
D	Data block	(0 bis 511)	Byte address	BY, W, DW

If the size of the data field is defined as a linear area the data field number must be set to 255.

8.5 Additional Functions

In addition to the random write and read access to PLC variables, a memory area comprising 12 bytes is specified in the application description as poll area.

Only marginal conditions regarding this memory area are that the PLC must be able to access in bit-mode and the terminal in word-mode and the memory area must be contiguous. The location of this memory area can be specified in the application description.

The data area comprises a maximum of 6 words or 12 bytes.
Example: The cyclic data area is set to DW21 in the programming system

Word address	DW	High-byte	Low-byte
Word address +0	DW21	Write coordination byte	Reserved
Word address +1	DW22	Message channel high-byte	Message channel low-byte
Word address +2	DW23	Function key LED 1 to 4	LED 5 to 8
Word address +3	DW24	Function key LED 9 to 12	LED 13 to 16
Word address +4	DW25	Function key LED 17 to 20	LED 21 to 24
Word address +5	DW26	Function key LED 25 to 28	LED 29 to 32

8.6 Error Messages

Code 0

Subcode

0 NO_ERROR.....Error-free processing

Code 1

Subcode

1 E_SLAVE_NOT_READY .....Error originating in level 1 and level 2
2 E_PROTOKOL .....Sequence of the packets
3 E_FRAME.....Character frame error
4 E_TIMEOUT.....Timeout error
5 E_CRC_BCC.....CRC error
6 E_PARITY.....Parity error
7 E_SEND_ABORT .....Abort send process
8 E_REC_ABORT.....Abort receive process
9 E_BUF_SIZE.....Insufficient cyclic buffer
10 E_NO_DEFINE.....No cyclic data defined

12 E_DEFINE .....Cyclic data already defined

15 E_NO_PROTOCOL .....Selected protocol is not supported
16 E_OVERRUN.....Receive buffer overrun

40 E_SYS_ADDRESS .....Illegal system variable

Bosch-specific error messages

Subcode

50 E_QUITTUNG_START .....No communication setup
51 E_QUITTUNG_OPEN .....Incorrect acknowledge signal during communication setup
52 E_QUITTUNG_DATA.....Incorrect acknowledge signal to transmitted information block

Possible errors:

- Check the settings for block check. The PU always uses LRC8. The first peripheral user gives the following block check.

53 E_NO_RESPONSE_WRONG_ CHAR . . . . . .No response message
54 E_TIMEOUT_NO_RESPONSE . . . .Timeout - no response message
55 E_TIMEOUT_BLOCKZEIT . . . . . .Timeout - block time period has been exceeded
56 E_TIMEOUT_QUIT_RESPONSE . .Timeout - no acknowledge signal
57 E_ABBRUCH_SPS . . . . . .EOT -PLC abort

Code 2

Subcode

58 E_RECEIVE_COUNT. . . . . .Number of received data is incorrect

Possible errors:

- Check if in the mask where the error occurs a variable with odd number of bytes will be read by a word or double-word address.

See module manual

Code 3

Subcode

1 The requested module is not available

16 Module is not accessible

35 The access to this address field is not allowed
36 The address field is secured by a user
37 Timer must not be written to
38 Module number to high
39 Module not available
40 Module is to small

147 Flag area overrun (CL200 only)
Possible errors:

- Flag area defined out of range of MB0 to MB191

Code 4

Subcode

32 The requested data type (Command Code) is not known by the PST

33 Protocol flag not known by the PST

35 The given coordination flag is not known by the PST

37 Parameter identifier in the telegram doesn't fit to the specified parameters

38 Length of block and topic number of data are different

40 Type of telegram unknown

41 Type of command unknown

58 Start address doesn't fit to the type of operand (Word at odd address)
Possible errors:
- Module R500 faulty

59 Start address defined outside the address area

60 Invalid parameter for specified command

61 Invalid type of operand

64 The PST hasn't received an identification telegram, yet

99 The given length of data is greater than the requested data area

210 Coordination flag is locked

SECTION 9
Allen Bradley - Interfacing via DF1

9.1 General Information

The operating terminals allow for a simple connection to the Allen-Bradley PLCs. The operating terminal is connected to the interface of the respective PLC-module. The data communication on the interface is handled by the DF1 protocol. Communication is possible with any system implementing the DF1 protocol (e.g.: MicroLogix 1000-Series, SLC500 with SLC5/03, SLC500 with SLC5/04, PLC5 and PLC5/250).

The software components of the system are fully adapted to the parameters and marginal conditions of the PU interface.

This offers the user the following advantages:

- Random write and read access to the defined data file.
- The operating terminal automatically polls the freely definable data areas for cyclic data.
- No configuration required within the PLC.
- The PU protocol is handled entirely by the firmware of the PLC. A PLC program (function blocks, etc.) in the PLC is not required for the handling of the communication.
- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission.
- The parameters of the interface SER1 are assigned in the programming software in a protocol-specific manner and are stored in the Application description. Modifying of the parameters is also possible in the setup mask or each other I/O mask of the terminal at any time.
- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC programm.

9.2 Technical Description

The interfacing of the operating terminal to the Allen-Bradley PLCs is effected in full-duplex mode.

The DF1 protocol allows random read and write access to the defined integer number file in either bit or word-mode.

Before individual bits of a word can be accessed for a write operation, a read access to the corresponding word must occur. After having modified the respective bit as desired, the word is transmitted to the PLC. Therefore, special care must be taken when accessing bits to ensure that the **terminal and the PLC** do not modify individual bits within a word at the same time.

9.3 Protocol Parameters Allen Bradley

The operating terminal adapts to the default parameters of the PU-interface. It is therefore **not** necessary to modify the interface parameters in the PLC.
To ensure proper communication, the parameters must **not** be altered.

Baud rate:	600, 1200, 2400, 4800, **9600**, 19200, 38400, 375000, 500000 Baud
Parity:	none, **even**, odd
Data bits:	5, 6, 7, **8**
Stopbits:	**1**, 1.5, 2
Handshake:	**no handshake**, hardware, software

9.3.1 Parameter Controller Type

Allows entering of the controller type that is being used.
Interfacing is possible to a SLC500 - 5/03 and a PLC 5 only.

9.3.2 Parameter Block Check

The DF1 protocol permits optionally use of the block check method CRC16 or BCC.

9.4 Data Type Structure

a) Alphanumerical Texts
Are stored in the memory byte for byte in ascending address order.

b) Binary Variables with a Length of 1, 2 or 4 Bytes
Data with a length of 2 bytes are interpreted in the PLC-conformal byte order for words.
Data with a length of 4 bytes are interpreted in the PLC-conformal byte order for long words.

c) Floating point numbers
The data will be interpreted in the IEEE-Floating point order.

9.4.1 Data Types

Interfacings to a PLC 5 permit defining of a data file number (9 to 254) for each variable in the variable list. For each data file number defined, a respective data file must be created in the PLC 5.

The following types of data are available for direct access:

B	Bit access	The address is a word address
W	Word access	The address is a word address
DW	Double-word access	The address is a word address

9.5 Additional Functions

In addition to the random write and read access to PLC variables, a memory area comprising 12 bytes is specified in the mask definition as poll area. The location of this memory area is specified in the mask definition.

Only marginal conditions regarding this memory area is that the PLC must be able to access in bit-mode and the terminal in word-mode, and the memory area must be contiguous. The address of this data area is defined in the application description.

The data area comprises a maximum of 6 words.

Example: The cyclic data area is set to N9:10 in the programming system

Word address		**High-byte**	**Low-byte**
Word address +0	N9:10	Write coordination byte	Reserved
Word address +1	N9:11	Message channel high-byte	Message channel low-byte
Word address +2	N9:12	Function key LED 1 to 4	LED 5 to 8
Word address +3	N9:13	Function key LED 9 to 12	LED 13 to 16
Word address +4	N9:14	Function key LED 17 to 20	LED 21 to 24
Word address +5	N9:15	Function key LED 25 to 28	LED 29 to 32

9.7 Error Messages

Code 0

Subcode

0

Code 1

Subcode

1	E_SLAVE_NOT_READY	Slave not ready
2	E_PROTOKOL	Sequence of the packets
3	E_FRAME	Character frame error
4	E_TIMEOUT	Timeout error
5	E_CRC_BCC	CRC error
6	E_PARITY	Parity error
7	E_SEND_ABORT	Abort send process
8	E_REC_ABORT	Abort receive process
9	E_BUF_SIZE	Insufficient cyclic buffer
10	E_NO_DEFINE	No cyclic data defined
12	E_DEFINE	Cyclic data already defined
15	E_NO_PROTOCOL	Selected protocol is not supported
16	E_OVERRUN	Receive buffer overrun
40	E_SYS_ADDRESS	Undefined system variable
50	E_QUIT_DATA	No acknowledge signal to order message
51	E_QUIT_DATA_NAK	Negative acknowledge signal to order message
52	E_WRONG_QUIT	Incorrect character for acknowledge signal
53	E_WRONG_REPLY	Reply can not be interpreted
55	E_TIMEOUT_ORDER	Timeout no response message
56	E_TIMEOUT_REPLY	Timeout no response message

Code 2

Subcode

58 E_RECEIVE_COUNT ..... Number of received data is incorrect

Possible errors:

- Check if in the mask where the error occurs a variable with odd number of bytes will be read by a word or double-word address.

Code 3 Error from the PLC

Subcode

10	Error in the order message
50	Access on address in the PLC that is not permitted

SECTION 10
Mitsubishi MelsecA

10.1 General Information

The operating terminals allow for a simple connection to every Mitsubishi PLC of the An / AnA series.

The operating terminal is connected to the **communications module AJ71C24**.

The software components of the system are fully adapted to the parameters and marginal conditions of the interface.

This offers the user the following advantages:

- Random write and read access to any data within the PLC. Data of existing PLC programs can be displayed and modified directly in the operating terminal. It is not necessary to adapt the PLC program to the operating terminal in any respect since it is not required that communications data be stored in a specified address area or data type area.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- No configuration is required in the PLC.

- The protocol is handled entirely by the operating system of the communications module. A PLC program in the PLC is not required for the handling of the communication.

- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission. A noise-immune interface hardware in accordance with the RS485 standard permits the application even in a harsh industrial environment.

- The interface parameters are stored in the application description. Modifying of the parameters is possible in the setup mask or in each other I/O mask of the terminal at any time.

- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program (e.g. M3 corresponds to flag 3).

10.2 Technical Description

The interfacing of the operating terminal to the Mitsubishi A - PLC is effected by means of the protocol type ”**Protocol 1**” of the communications module.

This allows random read and write access to all PLC data. Any bit-operands can be accessed in bit-mode or word mode. The size of the address area depends on the respective PLC.

10.3 Protocol Parameters Mitsubishi MelsecA

The parameters of the interface SER1 are set to the following values:

Baud rate:	300, 600,1200, 2400, 4800, 9600, **19200**, 38400, 357000, 500000 Baud
Parity:	none, **even**, odd
Data bits:	5, 6, **7**, 8
Stopbits:	**1**, 1.5, 2
Handshake:	**no handshake**, hardware, software

The default values of the programming system are printed in **bold**.

10.3.1 Parameter Maximum Waiting Time for Response

This timer indicates the length of time that the operating terminal (master) will wait for the response from the PLC (slave).
Permitted values are in the range of 0 ms to 65535 ms.
The default value is 500 ms.

10.3.2 Parameter Delay Until Connection Setup

Specifies the period of time that the terminal allows to elapse after an unsuccessful attempt to establish communication and before making another attempt.
Permitted values are in the range of 5000 ms to 255000 ms.
The default value is 10000 ms.

10.4 Parameters of the AJ71C24

10.4.1 Mode Selection Switch

The mode selection switch permits selection of the protocol format corresponding to the respective interfaces.

The protocol "**Protocol 1**" must be selected for the RS422 / RS485 interface to the terminal.

10.4.2 Station Number

The station number is used to specify the various communication modules within a multipoint connection. The specified station number must coincide with the specification in the variable list.

It is not necessary to comply with a specific sequence when assigning the station numbers, i.e. the numbering does not need to be sequential or contiguous.

Valid values for station numbers are: 0 to 31

10.4.3 DIP Switches

The following interface settings must be defined:

RS422 / RS485 as forward channel, to which the terminal is connected (if available)
Data length 7 bits
19200 Baud
Parity check, activated
Even parity
1 stop bit
Check sum, activated
Write access, permitted
Resistance for Termination Transmit-line, activated (if available)
Resistance for Termination Receive-line, activated (if available)

The functions of the DIP switches vary with the respective AJ71C24 model. Therefore, please consult the manual of the respective communications processor type for information on the specific functions of the switches.

10.5 Data Types

Bit operands with access in bit or word-mode

X	Inputs	hexadecimal input
Y	Outputs	hexadecimal input
M	Flags	decimal input
L	Latch flags	decimal input
S	Step flags	decimal input
B	Link flags	hexadecimal input
F	Error flags	decimal input
TS	Timer contact	decimal input
TC	Timer coil	decimal input
CS	Counter contact	decimal input
CC	Counter coil	decimal input

Word operands with access in word or double word-mode

TN	Timer actual value	decimal input
CN	Counter actual value	decimal input
D	Data register	decimal input
W	Link register	hexadecimal input

10.6 Additional Functions

In addition to the random write and read access to PLC variables, a memory area comprising 12 bytes is specified in the application description as cyclic data area.

This data area must be assigned to an address which can be accessed in word-mode. Therefore, the size of the poll area should comprise an equal number of bytes.

Word-structured Memory Mapping

The data area comprises a maximum of 12 bytes.

Example: Cyclic data starting with D1

Word address	**D**	**High-byte**	**Low-byte**
Word address +0	D1	Write coordination byte	Reserved
Word address +1	D2	Message channel high-byte	Message channel low-byte
Word address +2	D3	Function Key LED 1 to 4	LED 5 to 8
Word address +3	D4	Function Key LED 9 to 12	LED 13 to 16
Word address +4	D5	Function Key LED 17 to 20	LED 21 to 24
Word address +5	D6	Function Key LED 25 to 28	LED 29 to 32

10.7 Error Messages

Code
- 1 E_SLAVE_NOT_READY .....Slave not ready or defective cable or incorrect station number
- 2 E_PROTOKOL .....Sequence of the packets
- 3 E_FRAME .....Character frame error
- 5 E_CRC_BCC.....CRC error
- 6 E_PARITY.....Parity error
- 9 E_BUF_SIZE.....Insufficient cyclic buffer
- 10 E_NO_DEFINE.....No cyclic data defined
- 16 E_OVERRUN.....Receive buffer overrun
- 40 E_SYS_ADDRESS .....Incorrect system variable

Messages from the AJ71C24

- 50 No function in the RUN-mode.
  Possible errors
  - Set the respective DIP switch to the ON-position and restart or modifying of the parameters only when CPU in the STOP-mode.
- 51 Parity error
  Possible errors:
  - Set DIP-switch to correct parity setting (even parity).
- 52 Check sum error
  Possible errors:
  - Set DIP switch to correct check sum setting.

53 Protocol error
Possible errors:
- Mode switch does not correspond to the required protocol format. Correct mode switch.

54 Run-time error
Possible errors:
- The data received do not conform with the number of stop bits selected. Correct DIP switch.

55 Data overflow
Possible errors:
- New data were transmitted before the preceding transmission was completed.

56 Character set error
Possible errors:
- Invalid operand address or requested service not available.

57 Character error
Possible errors:
- Characters transmitted do not correspond to the valid character set.

58 Faulty access to CPU
Possible errors:
- This PLC type can not be used with the AJ71C24.

66 Assigned numbers are incorrect
Possible errors:
- The PLC-CPU number does not correspond to the value FF (hex).

67 Incorrect mode
Possible errors:
- Faulty communication between AJ71C24 and the CPU.

68 Incorrect assignment of the special module
Possible errors:
- An interactive special module comprising its own buffer was configured improperly in the system.

69 Incorrect step number in the PLC program
Possible errors:
- A step control instruction is outside of the area parameterized for the PLC-CPU or the sub-program to be executed is invalid.

74 Remote error
Possible errors:
- Remote RUN/STOP mode not possible.

82 Data-Link error
Possible errors:
- An attempt was made to access a station to which the communication has already been disconnected.

83 Data bus error at the special module
Possible errors:
- The memory of a special module can not be accessed.

SECTION 11
AEG - PU (KS-Functions)

11.1 General Information

The operating terminals allow for a simple connection to PLCs of the AEG Modicon series.
The operating terminal is connected to the interface of the ALU.
The data communication on the interface is handled by a protocol which is based on the KS-functions of the Modicon series.
Communication is possible with any system implementing the KS-functions based on the Modicon series:

AEG-MIKRO	PU-interface / RS232
AEG-120	PU-interface / RS232
AEG-250	PU-interface / RS232

The software components of the system are fully adapted to the parameters and marginal conditions of the interface.

This offers the user the following advantages:

- Random write and read access to any data within the PLC. Data of existing PLC programs can be displayed and modified directly in the operating terminal. It is not necessary to adapt the PLC program to the operating terminal in any respect. It is not required that communications data be stored in a specified address area or data type area.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- No configuration required within the PLC.

- The protocol is handled entirely by the firmware of the PLC. A PLC program (function blocks, etc.) in the PLC is not required for the handling of the communication.

- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission. An electrically isolated, noise-immune interface hardware in accordance with the RS232 standard permits the application even in a harsh industrial environment.

- The parameters of the interface SER1 are assigned in the programming software in a protocol-specific manner and are stored in the application description. Modifying of the parameters is also possible in the setup mask or any other I/O mask of the terminal at any time.

- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program.

11.2 Technical Description

With the aid of the KS-functions, all PLC data can be accessed in read and write-mode as well as in bit, byte, word and double word-mode. The size of the address area depends on the respective PLC and its configuration.

11.3 Protocol Parameters AEG-KS

The operating terminal adapts to the default parameters of the interface. It is therefore **not** necessary to modify the interface parameters in the PLC.
To ensure proper communication, the parameters must **not** be altered.

Baud rate:	**9600** Baud
Parity:	**even**
Data bits:	**8**
Stopbits:	**1**
Handshake:	**no handshake**

11.4 Data Type Structure

a) Alphanumerical Texts
Are stored in the memory byte for byte in ascending address order.

b) Counter
The count value is interpreted in the binary format. The maximum value is one word.

c) Timer
The time value is interpreted in the binary format. The maximum value is one word.

d) Binary Variables with a Length of 1, 2 or 4 Bytes
Data with a length of 2 bytes are interpreted in the PLC-conformal byte order for words.
Data with a length of 4 bytes are interpreted in the PLC-conformal byte order for double words.

e) Floating Point Numbers
Floating point numbers are interpreted in the IEEE floating point format.

11.4.1 Data Types

Direct accessing of the following data types is possible:

E	input bit
EB	input byte
EW	input word
ED	input double word
A	output
AB	output byte
AW	output word
AD	output double word
M	flag
MB	flag byte
MW	flag word
MD	flag double word
MG	flag floating point

The size of each area is governed by the configuration of the PLC.

T	timer-status
TI	timer actual value
TS	timer setpoint value
Z	counter
ZI	counter actual value
ZS	counter setpoint value

11.5 Additional Functions

In addition to the random write and read access to PLC variables, a memory area comprising 12 bytes is specified in the application description as poll area.
Only marginal conditions regarding this memory area is that the PLC must be able to access in bit-mode and the terminal in word-mode and that the memory area must be contiguous. The location of this memory area can be specified in the application description.

The data area has a maximum size of 6 words or 12 bytes.
It can optionally be located in a byte or word-structured area.

Byte-structured Memory Mapping

The data area comprises a maximum of 11 bytes

Example: The cyclic poll area is set to MB21 in the programming system.

Byte address	MB	Description
Byte address +0	MB21	Write Coordination Byte
Byte address +1	MB22	Message Channel Low-byte
Byte address +2	MB23	Message Channel High-byte
Byte address +3	MB24	Function Key LED 1 to 4
Byte address +4	MB25	Function Key LED 5 to 8
Byte address +5	MB26	Function Key LED 9 to 12
Byte address +6	MB27	Function Key LED 13 to 16
Byte address +7	MB28	Function Key LED 17 to 20
Byte address +8	MB29	Function Key LED 21 to 24
Byte address +9	MB30	Function Key LED 25 to 28
Byte address +10	MB31	Function Key LED 29 to 32

Word-structured Memory Mapping

The data area comprises a maximum of 12 bytes (6 words).

Example: The cyclic data area is set to MW10 in the programming system.

Word address	MW	High-byte	Low-byte
Word address +0	MW10	Write coordination byte	Reserved
Word address +1	MW11	Message channel high-byte	Message channel low-byte
Word address +2	MW12	Function key LED 1 to 4	LED 5 to 8
Word address +3	MW13	Function key LED 9 to 12	LED 13 to 16
Word address +4	MW14	Function key LED 17 to 20	LED 21 to 24
Word address +5	MW15	Function key LED 25 to 28	LED 29 to 32

11.6 Error Messages

Code 1 Error from the data transmission-layer

Subcode

1	E_SLAVE_NOT_READY	Slave not ready
2	E_PROTOKOL	Sequence of the packets
3	E_FRAME	Character frame error
4	E_TIMEOUT	Timeout error
5	E_CRC_BCC	CRC error
6	E_PARITY	Parity error
7	E_SEND_ABORT	Abort send process
8	E_REC_ABORT	Abort receive process
9	E_BUF_SIZE	Insufficient cyclic buffer
10	E_NO_DEFINE	No cyclic data defined
12	E_DEFINE	Cyclic data already defined
15	E_NO_PROTOCOL	Selected protocol is not supported
16	E_OVERRUN	Receive buffer overrun
40	E_SYS_ADDRESS	Undefined system variable

AEG - Specific Error Messages:

Code 1 Error from the data transmission-layer

Subcode

50 E_NO_POLL_QUIT .....No acknowledge signal upon polling

51 E_NO_TELE_QUIT.....No acknowledge signal upon order-message

52 E_NO_TELE_POLL_QUITT .....No acknowledge signal upon tele-polling

54 E_POLL_TIMEOUT.....Timeout - no response upon polling

55 E_TELE_TIMEOUT.....Timeout - no response message

56 E_NO_RESP_QUIT_QUIT .....Timeout - no response upon response-acknowledge signal

Code 3 Error from the PLC

Subcode

05 Wrong type of operand

06 Valid area exceeded

SECTION 12
JETTER PASE/PCOM5

12.1 General Information

The operating terminals allow for a simple connection to every JETTER PASE-E and JETTER PASE-Mikro controller.

The operating terminal is connected to the RS232 programming unit interface.

The software components of the system are fully adapted to the parameters and marginal conditions of the interface.

This offers the user the following advantages:

- Random read and write access to any data within the PLC. Data of existing PLC programs can be displayed and modified directly in the operating terminal. It is not necessary to adapt the PLC program in any respect since it is not required that communications data be stored in a specified address area or data type area.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- No configuration required within the PLC.

- The protocol is handled entirely by the operating system of the communications module. A PLC program is not required for the handling of the communication.

- The parameters of the interface SER1 are stored in the application description. Modifying of these parameters is possible in the setup mask or each other I/O mask of the terminal at any time.

- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program (e.g. M3 = flag 3).

12.2 Technical Description

The interfacing of the operating terminal to JETTER PASE-E and JETTER PASE-Mikro controllers is effected by means of the protocol PASE or PCOM5.

Both the PASE and the PCOM5 protocol permit access to one variable at a time only. To allow access to linear blocks of data, the communications driver of the operating terminal automatically segments block-oriented access procedures into separate accesses to the PLC.

The inputs of the PLC can be accessed via the protocol in the read-mode only.

The data type 32-Bit-Real-Register (Register address 8960 to 9215) is only available with the JETTER PASE-E controller.

Access to alphanumeric text-variables is possible via the data type **T** (24-Bit-Text-Register) only. During such accesses, the terminal will not carry out any length information management or status information management, respectively. The text data are located in the 3-byte register in a flushed manner.

12.3 Protocol Parameters Jetter

The following parameters are set for the protocol **PASE/PASE-E**:

Baud rate:	300, 600, 1200, 2400, 4800, **9600**, 19200, 38400, 357000, 500000 Baud
Parity:	**none**, even, odd
Data bits:	5, 6, 7, **8**
Stop bits:	**1**, 1.5, 2
Handshake:	**no handshake**, hardware, software

The following parameters are set for the protocol **PCOM5 :**

Baud rate:	300, 600, 1200, 2400, 4800, **9600**, 19200, 38400, 357000, 500000 Baud
Parity:	none, **even**, odd
Data bits:	5, 6, 7, **8**
Stop bits:	**1**, 1.5, 2
Handshake:	**no handshake**, hardware, software

The default parameters of the programming system are printed in **bold**.

12.3.1 Parameter Maximum Waiting Time for Response

This timer indicates the length of time that the operating terminal will wait for the response from the PLC.
Permitted values are in the range of 0 ms to 65535 ms.
The default value is 500 ms.

12.3.2 Parameter Delay Until Connection Setup

Specifies the period of time that the terminal allows to elapse after an unsuccessful attempt to establish the communication and before making another attempt.
Permitted values are in the range of 500 ms to 65535 ms.
The default value is 1000 ms.

12.3.3 Parameter Jetter CPU-Type

Depending on the configuration of the PLC a CPU-type must be selected for the protocol.
The following CPU-types can be selected:

- PASE-Micro
- PASE-E
- Delta
- Nano

12.3.4 Parameter Protocol-Type

As protocol-type the entries PASE-E or PCOM5 can be selected.

12.4 Data Types

Bit Data Types

E input
A output
M flag

Word Data Types

R 24-Bit-Signed-Integer-Register or 32-Bit-Real-Register
T 24-Bit-Text-Register

12.5 Address Input (Number of the Variable)

The upper limit of the address input is dependent on the respective PLC. Therefore, please consult the PLC manual. Listed below are explanations of the lower limits or of the differences between the various address ranges, respectively:

Input and Output

PASE-Mikro	E 1 to E 16	A 1 to A 16
PASE-E	E 101 to E 116	A 101 to A 116
	E 201 to E 216	A 201 to A 216

etc. depending on the number of inputs and outputs

Flag

M0 to M upper limit (depends on PLC)

Register

24-Bit-Signed-Integer-Register

R 0 to R 8959

24-Bit-Text-Register

T 0 to T 8959

32-Bit-IEEE-Floating-Point-Register (Only available with the PASE-E PLC)

R 8960 to R 9215

12.6 Additional Functions

Cyclic Poll Area

In addition to the random read and write access to PLC variables, a memory area comprising 12 bytes is specified in the application description as cyclic poll area.

This data area must be referenced to a **24-Bit-Integer-Register-Variable**. This variable represents the starting address of the data area. To avoid problems with the signs, the use of the 24-Bit-Integer-Registers is limited to the lower 2 bytes. When defining the size in the software, it is only necessary to specify the number of bytes actually used. The software will then increase this size by the number of bytes that are not used.

Memory Mapping of the Word-Structured Poll Area

Example: Cyclic data starting with register 100, with a size of 12 bytes having been defined

High-Byte	Mid-Byte	Low-Byte	
	KBS	Reserved	Register 100
	Message-high	Message-low	Register 101
	LED 1 to 4	LED 5 to 8	Register 102
	LED 9 to 12	LED 13 to 16	Register 103
	LED 17 to 20	LED 21 to 24	Register 104
	LED 25 to 28	LED 29 to 32	Register 105
			etc.

The high-byte is not used

Parallel Message System

With the parallel message system, individual bits in the controller are assigned status messages.

This data area must be referenced to a **24-Bit-Integer-Register-Variable**. This variable represents the starting address of the data area. To avoid problems with the signs, the use of the 24-Bit-Integer-Registers is limited to the lower 2 bytes. When defining the size in the programming software, it is only necessary to specify the number of bytes actually used. The software will then increase this size by the number of bytes that are not used.

Memory Mapping of the Parallel Message System

Example: Status messages starting with register 200, with a size of 6 bytes having been defined

High-Byte	Mid-Byte		Low-Byte		
	16	8	7	1	Register 200
	32	24	23	17	Register 201
	48	40	39	33	Register 202
					etc.

The high-byte is not used

12.7 Error Messages

Code

1	E_SLAVE_NOT_READY	Slave not ready or defective cable
2	E_PROTOKOL	Characters received do not conform with protocol
3	E_FRAME	Character frame error
10	E_NO_DEFINE	No cyclic data defined
16	E_OVERRUN	Receive buffer overrun
40	E_SYS_ADDRESS	Incorrect system variable
50	E_WRITE_TO_INPUT	Write access to inputs not permissible
51	E_NO_REAL_TYPE	Real-registers are not permissible with PASE-Mikro

PCOM5-specific error messages

Code

52	E_NOT_STX	No STX (0xDA) at beginning of telegram
53	E_STATUS_PARA	Invalid parameter in telegram (status bit 1)
54	E_STATUS_COMAND	Invalid command in telegram (status bit 2)
55	E_STATUS_TIMEOUT	Timeout (status bit 3)
56	E_STATUS_ERROR	General error in status (status bit 4)
57	E_RECV_ERROR	Error on receiving the telegram (status bit 6)

SECTION 13
SUCOM1 - PS4-201

13.1 General Information

The operating terminals allow for a simple connection to the Klöckner Moeller PLCs PS 4-201.

The operating terminal is connected to the PU interface of the PLC. The data traffic on the interface is handled by the SUCOM1 protocol.

The software components of the system are fully adapted to the parameters and marginal conditions of the SUCOM1 protocol.

This offers the user the following advantages:

- Random read and write access to all flags within the PLC. Data of existing PLC programs can be displayed and modified directly in the operating terminal. It is not necessary to adapt the PLC program to the operating terminal in any respect because it is not required that communications data are stored in a specified address area or data type area.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- No additional configuration effort required within the PLC.

- Minimal increase of the cycle time of the PLC.

- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission.

- The parameters of the interface SER1 are stored in the application description. Modifying of these parameters is possible in the setup mask or each other I/O mask of the terminal at any time.

- The programming system provides a maximum of support to the user in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program.

13.2 Technical Description

The connection of the operating terminal to the Klöckner Moeller PLC is effected by means of the SUCOM1 programming protocol.
The SUCOM1 protocol allows random read and write access to all flags of the PLC. All data types can also be accessed in bit-mode. The size of the address area depends on the respective PLC.

A read access to the addressed word must occur, before individual bits can be accessed for a write operation. During these accesses, care must be taken to ensure that neither the terminal nor the PLC modify individual bits of the same word.

13.3 Protocol Parameters SUCOM1 PS4-201

The parameters of the interface SER1 are set to the following values:

Baud rate:	300, 600, 1200, 2400, 4800, **9600**, 19200, 38400, 375000, 500000 Baud
Data bits:	5, 6, 7, **8**
Stop bits:	1, 1.5, **2**
Parity:	**none**, even, odd
Handshake:	**no handshake**, hardware, software

The default parameters of the programming system are printed in **bold**.

13.4 Data Types

M	flag bits	(bit access)
MB	flag bytes	(byte access)
MW	flag word	(word access)
PS	process status word	(word access)
DS	diagnostic word	(word access)
DZ	diagnostic counter	(word access)

13.5 Additional Functions

In addition to the random read and write access to PLC variables, a memory area comprising 12 bytes is defined in the mask definition as cyclic poll area.

Only marginal conditions regarding this memory area are that the PLC must be able to access in bit-mode and the memory area must be contiguous. The starting address of this memory area must be specified in the mask definition and must be located on a word boundary.

Word-structured Memory Mapping

The data area comprises a maximum of 6 words.

Example: Cyclic data area on MW10

MW	**High-byte**	**Low-byte**
MW10	Write coordination byte	Reserved
MW12	Message channel high-byte	Message channel low-byte
MW14	LED 1 to 4	LED 5 to 8
MW16	LED 9 to 12	LED 13 to 16
MW18	LED 17 to 20	LED 21 to 24
MW20	LED 25 to 28	LED 29 to 32

13.6 Error Messages

Code	1	E_SLAVE_NOT_READY	Slave not ready
	2	E_PROTOKOL	Sequence of the packets
	3	E_FRAME	Character frame error
	4	E_TIMEOUT	Timeout error
	5	E_CRC_BCC	CRC error
	6	E_PARITY	Parity error
	7	E_SEND_ABORT	Abort send process
	8	E_REC_ABORT	Abort receive process
	9	E_BUF_SIZE	Insufficient cyclic buffer
	10	E_NO_DEFINE	No cyclic data defined
	12	E_DEFINE	Cyclic data already defined
	15	E_NO_PROTOCOL	Selected protocol is not supported
	16	E_OVERRUN	Receive buffer overrun
	17	E_NAK	NAK from the PLC
	40	E_SYS_ADDRESS	Undefined system variable

SUCOM1 Specific Error Messages

Code	50	E_QUITTUNG_COMM	No acknowledge signal during the connection setup
	51	E_QUITTUNG_DATA	No acknowledge signal has been received after sending an information block
	52	E_NO_DATA	No data upon request

Error in the Data Section

Code	53	E_RECEIVE_COUNT	Number of bytes received does not correspond to the number of bytes requested
	54	E_NOT_READY	No STX upon receipt of data block

SECTION 14
Idec Micro3

14.1 General Information

The operating terminals can be easily connected to Idec Micro3 controllers.
The programming interface is used to connect the operating terminal.

Data traffic on the interface is handled by the 1:N protocol.
The operating terminal is capable of addressing several controllers simultaneously.

The software components of the system are fully adapted to the parameters and marginal conditions of the programming unit (PU) interface.

This offers the user the following advantages:

- Random read and write access to all data within the PLC.
- The operating terminal automatically polls the freely definable data areas for cyclic data.
- No configuration effort required within the PLC.
- The PU protocol is handled entirely by the operating system of the controller. A PLC program (function blocks, etc.) in the PLC is not required for communication handling.
- The protocol provides error control. Transmission errors are detected and, if possible, eliminated by repeating the transmission. An electrically isolated, noise-immune interface hardware in accordance with the RS485 half-duplex standard permits the application even in a harsh industrial environment.
- The parameters of the interface SER1 are assigned in the programming software in a protocol-specific manner and are stored in the application description. Modifying of the parameters is also possible in the setup mask or each other I/O mask of the terminal at any time.
- The programming system provides a maximum of support to the operator in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program.

14.2 Technical Description

The 1:N protocol is used to connect the operating terminal to Idec Micro3 controllers.

This protocol allows random read and write access - in bit, byte and word-mode - to all PLC data.

14.3 Protocol Parameters IDEC Micro 3

The operating terminal adapts to the default parameters of the PU interface. It is therefore **not** necessary to adapt the interface parameters in the PLC.
To ensure proper connection, do **not** modify the parameters.

Baud rate:	300, 600, 1200, 2400, 4800, **9600**, 19200, 38400, 375000, 500000 Baud
Parity:	none, odd, **even**
Data bits:	5, 6, **7**, 8
Stop bits:	**1**, 1.5, 2
Handshake:	**no handshake**, hardware, software

14.3.1 Parameter Maximum Waiting Time for Response

This parameter indicates the length of time that the operating terminal waits for the response from the PLC. The valid range of values is between 10 ms and 2550 ms. The default value specified by the programming system is 500 ms.

14.3.2 Parameter Delay Until Connection Setup

This parameter is not evaluated.

14.4 Data Type Structure

a) Alphanumerical Texts
Are stored in the memory byte for byte in ascending address order.

b) Binary Variables with a Length of 1, 2 or 4 Bytes
Data with a length of 2 bytes are interpreted in the PLC-conforming byte order for words.
Data with a length of 4 bytes are interpreted in the PLC-conforming byte order for long-words.

c) Timer
For access to the timer, the corresponding mode of representation TIMER must be selected.

d) Counter
For access to the counter, the corresponding mode of representation COUNTER must be selected.

14.4.1 Data Types

Direct user access is possible to the following data types:

I	Input	Bit
IB	Input	Byte
IW	Input	Word
Q	Output	Bit
QB	Output	Byte
QW	Output	Word
M	Flag	Bit
MB	Flag	Byte
MW	Flag	Word
R	Register	Bit
RB	Register	Byte
RW	Register	Word
DW	Data Register	Word
TP	Timer PRESET value	Word Constant or data register no., read-only Use the data register to set the timer
TC	Timer CURRENT value	Word read-only
CP	Counter PRESET value	Word Constant or data register no., read-only Use the data register to set the counter
CC	Counter CURRENT value	Word read-only
HP	Highspeed Counter Preset	Double Word Constant or data register no., read-only Do not use in a table
HC	Highspeed Counter Current	Double Word read-only Do not use in a table

14.5 Additional Functions

In addition to the random read and write access to controller variables, a 12 byte memory area is specified in the mask definition as a poll area.

The only marginal conditions regarding this memory area are that the PLC must be able to access in bit-mode and the terminal in word-mode and the memory area must be contiguous. This memory area must reside in the data register (DW).

The maximum data area size is 6 words.

Example: The cyclic poll area is set to DW10 in the programming system.

Word address	**DW**	**High-byte**	**Low-byte**
Word address +0	DW10	Write coordination byte	Reserved
Word address +1	DW11	Message channel high-byte	Message channel low-byte
Word address +2	DW12	Function key LEDs 1 to 4	LEDs 5 to 8
Word address +3	DW13	Function key LEDs 9 to 12	LEDs 13 to 16
Word address +4	DW14	Function key LEDs 17 to 20	LEDs 21 to 24
Word address +5	DW15	Function key LEDs 25 to 28	LEDs 29 to 32

14.6 Error Messages

Code 1

Subcode

1	E_SLAVE_NOT_READY	Slave not ready
2	E_PROTOKOL	Sequence of the packets
3	E_FRAME	Protocol frame error
4	E_TIMEOUT	Timeout error
5	E_CRC_BCC	CRC error
6	E_PARITY	Parity error
7	E_SEND_ABORT	Send process aborted
8	E_REC_ABORT	Receive process aborted
9	E_BUF_SIZE	Insufficient cyclic buffer
10	E_NO_DEFINE	No cyclic data defined
12	E_DEFINE	Cyclic data already defined
15	E_NO_PROTOCOL	Selected protocol is not supported
16	E_OVERRUN	Receive buffer overrun
40	E_SYS_ADDRESS	Undefined system variable
50	E_RESPONSE_TIMEOUT	No response from PLC

Code 2

Subcode

58	E_RECEIVE_COUNT	Number of received data incorrect. Check if a variable with an odd number of bytes is read from a word or double-word address in the mask in which the error occurs.

Code 3 PLC error

Subcode

6	Data area error	Write-access to flag >= 290. Address does not exist in PLC
7	Illegal write-access to timer and counter	Writing to timer and counter not permitted.
10	Incorrect data coding in the protocol	

Code 17

Subcode

00	BCC error on the PLC-end	
01	Frame error on the PLC-end	
02	Parity error on the PLC-end	
03	Illegal command to PLC	Write-access to highspeed counter.
04	Procedure error on the PLC-end	

SECTION 15
ABB CS31

15.1 General Information

The operating terminals can be easily connected to ABB CS31 controllers.
The programming interface COM1 is used to connect the operating terminal.
Data traffic on the interface is handled by the ASCII plain text protocol.

The software components of the system are fully adapted to the parameters and marginal conditions of the programming unit (PU) interface.

This offers the user the following advantages:

- Random read and write access to all data within the PLC.
- The operating terminal automatically polls the freely definable data areas for cyclic data.
- No configuration effort required within the PLC.
- The PU protocol is handled entirely by the firmware of the PLC. A PLC program (function blocks, etc.) in the PLC is not required for communication handling.
- The interface parameters of the interface SER1 are defined in the programming software in a protocol-specific manner and are stored in the application description. The parameters can also be modified in the setup mask or any other I/O mask of the operating terminal at any time.
- The programming system offers the user a maximum of support in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used within the PLC program.

15.2 Technical Description

COM1 and the ASCII plain text protocol are used to connect the operating terminal to ABB CS31 controllers.

This protocol allows random read and write access - in bit, word and double-word mode - to all relevant PLC data.

15.3 Protocol Parameters ABB CS31

The interface parameters of the operating terminal are adapted to the default parameters of the PU interface.
To ensure proper connection, do **not** modify the parameters in the terminal.

Baud rate:	300, 600, 1200, 2400, 4800, **9600**, 19200, 38400, 357000, 500000 Baud
Parity:	**none**, even, odd
Data bits:	5, 6, 7, **8**
Stop bits:	**1**, 1.5, 2
Handshake:	No handshake, **hardware**, software

15.3.1 Parameter Maximum Waiting Time for Response

This parameter indicates the length of time that the operating terminal waits for the response from the PLC.
Permitted values are in the range of 1 ms to 65535 ms.
The default value is 1000 ms.

15.3.2 Parameter Delay Until Connection Setup

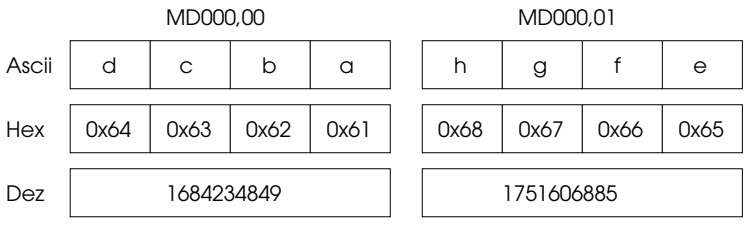
This parameter specifies the time the operating terminal waits before initiating communication.
Permitted values are in the range of 1000 ms to 65535 ms
The default value is 5000 ms.

15.4 Data Type Structure

a) Alphanumerical Texts
The texts are interpreted in accordance with the following byte order:

Word address

Double word address



b) Binary Variables with a Length of 2 or 4 Bytes
Data with a length of 2 bytes are interpreted in the PLC-conforming byte order for words.
Data with a length of 4 bytes are interpreted in the PLC-conforming byte order for long-words.

15.4.1 Data Types

Direct user access is possible to the following data types:

E	input	bit
EW	input	word
A	output	bit
AW	output	word
M	flag	bit
MW	flag	word
MD	flag	double word
KW	register	word
KD	register	double word

The addresses to the left and right of the comma are programmed in the programming software in separate columns.
When reading from the PLC, variables that are of the same type and are located in an area of up to 50 bytes are read as a block.

NOTE:
If the terminal accesses the local binary inputs E62.00 - E63.07 and local analog inputs EW06.00 - EW06.03, the PLC provides correct data only if the PLC program accesses an input of the corresponding group at least once.

15.5 Additional Functions

In addition to the random read and write access to PLC variables, a 12 byte memory area is specified in the application description as a poll area.

The only marginal conditions regarding this memory area are that the PLC must be able to access in bit-mode and the terminal in word-mode and the memory area must be contiguous.
This poll area must point to the **flag word area** (MW).

Example: The cyclic poll area is set to MW000,10 in the programming system.

Word address	**MW**	**High-byte**	**Low-byte**
Word address +0	MW000,10	Write coordination byte	Reserved
Word address +1	MW000,11	Message channel high-byte	Message channel low-byte
Word address +2	MW000,12	Function key LEDs 1 to 4	LEDs 5 to 8
Word address +3	MW000,13	Function key LEDs 9 to 12	LEDs 13 to 16
Word address +4	MW000,14	Function key LEDs 17 to 20	LEDs 21 to 24
Word address +5	MW000,15	Function key LEDs 25 to 28	LEDs 29 to 32

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Word address + 0	free	free	free	DDF	LM	PL	RQ	ED	free	free	free	free	free	free	free	free
Word address + 1	serial message channel high-byte								serial message channel low-byte							
Word address + 2	LED1 on/off	LED1 flashing	LED2 on/off	LED2 flashing	LED3 on/off	LED3 flashing	LED4 on/off	LED4 flashing	LED5 on/off	LED5 flashing	LED6 on/off	LED6 flashing	LED7 on/off	LED7 flashing	LED8 on/off	LED8 flashing
Word address + 3	LED9 on/off	LED9 flashing	LED10 on/off	LED10 flashing	LED11 on/off	LED11 flashing	LED12 on/off	LED12 flashing	LED13 on/off	LED13 flashing	LED14 on/off	LED14 flashing	LED15 on/off	LED15 flashing	LED16 on/off	LED16 flashing
Word address + 4	LED17 on/off	LED17 flashing	LED18 on/off	LED18 flashing	LED19 on/off	LED19 flashing	LED20 on/off	LED20 flashing	LED21 on/off	LED21 flashing	LED22 on/off	LED22 flashing	LED23 on/off	LED23 flashing	LED24 on/off	LED24 flashing
Word address + 5	LED25 on/off	LED25 flashing	LED26 on/off	LED26 flashing	LED27 on/off	LED27 flashing	LED28 on/off	LED28 flashing	LED29 on/off	LED29 flashing	LED30 on/off	LED30 flashing	LED31 on/off	LED31 flashing	LED32 on/off	LED32 flashing

NOTE:
To switch LED 1 on, the value -32768 (0x8000) must be entered in the address +2. However, the ASCII plain text protocol does not transfer this value.

SOLUTION:
Enter -32767 (0x8001) instead of -32768. This will set the flash bit of LED 8 which has no effect on the output.

15.6 Error Messages

Code 1

Subcode

1 E_SLAVE_NOT_READY .....Slave not ready
2 E_PROTOKOL .....Sequence of the packets
3 E_FRAME .....Protocol frame error
4 E_TIMEOUT .....Timeout error
5 E_CRC_BCC .....CRC error
6 E_PARITY .....Parity error
7 E_SEND_ABORT .....Send process aborted
8 E_REC_ABORT .....Receive process aborted
9 E_BUF_SIZE .....Insufficient cyclic buffer
10 E_NO_DEFINE .....No cyclic data defined

12 E_DEFINE .....Cyclic data already defined

15 E_NO_PROTOCOL .....Selected protocol is not supported
16 E_OVERRUN .....Receive buffer overrun

40 E_SYS_ADDRESS.....Undefined system variable

50 E_RESPONSE_TIMEOUT .....No response from PLC

Code 2

Subcode

52 E_RECEIVE_COUNT . . . . . .Number of received data incorrect.

Possible errors:

- Check if a variable with an odd number of bytes is read from a word or double-word address in the mask in which the error occurs.

54 E_ECHO . . . . . .Echo received and the transmitted data do not match.

70 E_RECEIVE_ERROR . . . . . .Unknown error from the PLC.

71 # WRONG VALUE

72 # WRONG INPUT

73 # NUMBER TOO LARGE

SECTION 16
ABB T200

16.1 General Information

Operating terminals can be easily connected to ABB T200 controllers.
The programming interface or the communication module is used to connect the operating terminal.
Data traffic on the interface is handled by the ASCII plain text protocol.

The software components of the system are fully adapted to the parameters and marginal conditions of the PLC interface.

This offers the user the following advantages:

- Random read and write access to all data within the PLC.

- The operating terminal automatically polls the freely definable data areas for cyclic data.

- No configuration effort required within the PLC.

- The protocol is handled entirely by the firmware of the PLC. A PLC program (function blocks, etc.) in the PLC is not required for communication handling.

- The interface parameters of the interface SER1 are defined in the programming software in a protocol-specific manner and are stored in the application description. The parameters can also be modified in the setup mask or any other I/O mask of the operating terminal at any time.

- The programming system offers the user a maximum of support in programming the operating terminal. The definitions (abbreviations) used here are identical with the definitions used in the PLC program.

16.2 Technical Description

The connection between operating terminals and ABB T200 controllers is established via the programming unit (PU) interface (RS232) as a point-to-point connection (1:1) or via the communication module (RS485) as a multipoint connection (1:N).

The protocol used for the connection is the „One-Way-Activation-Procedure" which is based on the „Communication Protocol" and operates on the master/slave principle.
The operating terminal acts as a master while the ABB T200 functions as a slave.

This protocol allows random read and write access - in bit, word and double-word mode - to all relevant PLC data.

16.3 Protocol Parameters ABB T200

The interface parameters of the operating terminal are adapted to the default parameters of the PU interface.

Baud rate:	300, 600, 1200, 2400, 4800, 9600, **19200**, 38400, 375000, 500000 baud
Parity:	none, **even**, odd
Data bits:	5, 6, **7**, 8
Stop bits:	**1**, 1.5, 2
Handshake:	no handshake, **hardware**, software

The default settings in the programming system are printed in **bold**.

16.3.1 Parameter Maximum Waiting Time for Response

This parameter indicates the length of time that the operating terminal (master) waits for the response from the PLC (slave).
Permitted values are in the range of 1 ms to 65535 ms.
The default value is 1000 ms.

16.3.2 Parameter Delay Until Connection Setup

This parameter specifies the time the operating terminal waits before initiating communication.
Permitted values are in the range of 100 ms to 25500 ms
The default value is 5000 ms.

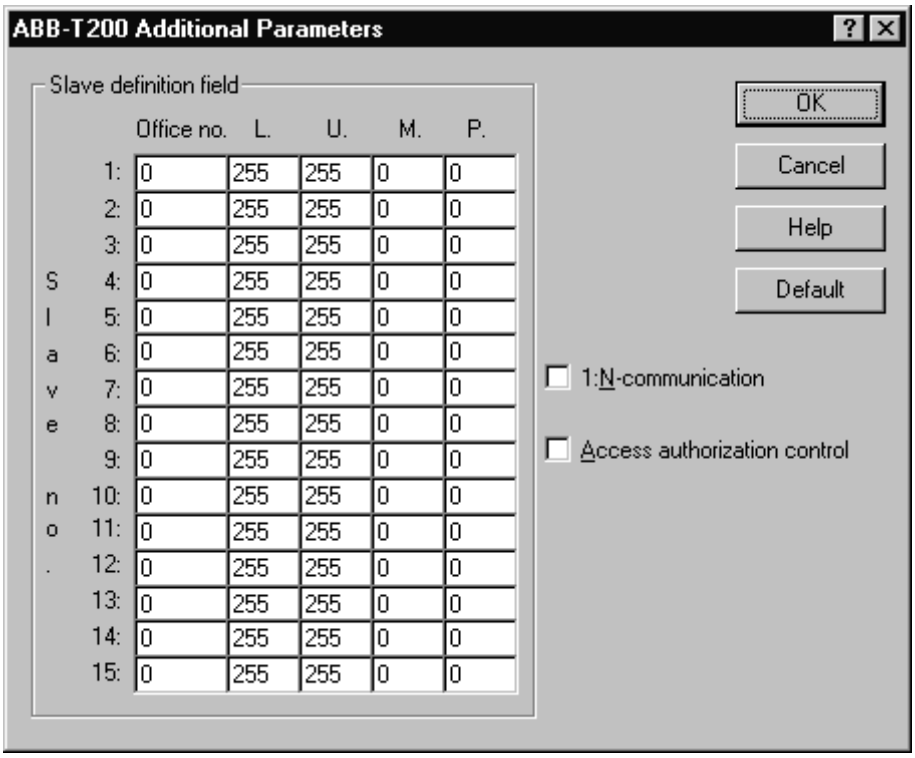
16.3.3 Additional Parameters

A network address is assigned to every variable by means of the slave number. This assignment is carried out in the programming software via the variable list. This address is used to exchange data between the operating terminal and controller.

Enter the corresponding information into the additional parameters to establish an assignment between the L.U.M.P. address and slave number. When using multipoint connections (1:N), also enter the office numbers.

If the check box **1:N Communication** is selected, the protocol also transfers the office numbers.

The check box **Access Authorization Control** ensures that the operating terminal is checked for an access authorization (whether it has or will receive this access authorization) prior to every access to the controller. The access authorization control feature is only required if more than four devices access the memory of the controller via the protocol.



The button Default can be used to restore the default values in the table.

16.4 Data Type Structure

a) Alphanumerical Texts

The texts are interpreted in accordance with the following byte order:

Word address

	MW000,00		MW000,01		MW000,02		MW000,03	
Ascii	b	a	d	c	f	e	h	g
Hex	0x62	0x61	0x64	0x63	0x66	0x65	0x68	0x67
Dez	25185		25699		26213		26727	

Double word address

	MD000,00				MD000,01			
Ascii	d	c	b	a	h	g	f	e
Hex	0x64	0x63	0x62	0x61	0x68	0x67	0x66	0x65
Dez	1684234849				1751606885			

16.4.1 Data Types

Direct user access is possible to the following data types:

E	input	bit	read/write
EW	input	word	read/write
ED	input	double word	read/write
E’	input	communication bit	read/write
EW’	input	communication word	read/write
ED’	input	communication double-word	read/write
A	output	bit	read/write
AW	output	word	read/write
AD	output	double word	read/write
A’	output	communication bit	read/write
AW’	output	communication word	read/write
AD’	output	communication double-word	read/write
M	flag	bit	read/write
MW	flag	word	read/write
MD	flag	double word	read/write
M’	flag	bit from bit/word area	read/write
MW’	flag	word from bit/word area	read/write
MD’	flag	double word from bit/word area	read/write
T	timer	status	read-only
TI	timer	actual value	read-only
Z	counter	status	read-only
ZI	counter	actual value	read-only

16.4.2 Range of Values

The range of values below applies to the following data types:

Data Type	Range of Values	Specific Characteristics
bit	00 and 01	none
word	0x7FF to 0x8001	The value 0x8000 is not permitted
double word	0x7FFFFFFF to 0x80000001	The value 0x80000000 is not permitted

16.4.3 Additional Functions

In addition to the random read and write access to PLC variables, a 12 byte memory area is specified in the application description as a poll area.

The only marginal conditions regarding this memory area are that the PLC must be able to access in bit-mode and the terminal in word-mode and the memory area must be contiguous.
This poll area must point to the **flag word area** (MW).

Example: The cyclic poll area is set to MW'000,10 in the programming system.

Word address	**MW'**	**High-byte**	**Low-byte**
Word address +0	MW'000,10	Write coordination byte	Reserved
Word address +1	MW'000,11	Message channel high-byte	Message channel low-byte
Word address +2	MW'000,12	Function key LEDs 1 to 4	LEDs 5 to 8
Word address +3	MW'000,13	Function key LEDs 9 to 12	LEDs 13 to 16
Word address +4	MW'000,14	Function key LEDs 17 to 20	LEDs 21 to 24
Word address +5	MW'000,15	Function key LEDs 25 to 28	LEDs 29 to 32

	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Word address + 0	free	free	free	DDF	LM	PL	RQ	ED	free	free	free	free	free	free	free	free
Word address + 1	serial message channel high-byte								serial message channel low-byte							
Word address + 2	LED1 on/off	LED1 flashing	LED2 on/off	LED2 flashing	LED3 on/off	LED3 flashing	LED4 on/off	LED4 flashing	LED5 on/off	LED5 flashing	LED6 on/off	LED6 flashing	LED7 on/off	LED7 flashing	LED8 on/off	LED8 flashing
Word address + 3	LED9 on/off	LED9 flashing	LED10 on/off	LED10 flashing	LED11 on/off	LED11 flashing	LED12 on/off	LED12 flashing	LED13 on/off	LED13 flashing	LED14 on/off	LED14 flashing	LED15 on/off	LED15 flashing	LED16 on/off	LED16 flashing
Word address + 4	LED17 on/off	LED17 flashing	LED18 on/off	LED18 flashing	LED19 on/off	LED19 flashing	LED20 on/off	LED20 flashing	LED21 on/off	LED21 flashing	LED22 on/off	LED22 flashing	LED23 on/off	LED23 flashing	LED24 on/off	LED24 flashing
Word address + 5	LED25 on/off	LED25 flashing	LED26 on/off	LED26 flashing	LED27 on/off	LED27 flashing	LED28 on/off	LED28 flashing	LED29 on/off	LED29 flashing	LED30 on/off	LED30 flashing	LED31 on/off	LED31 flashing	LED32 on/off	LED32 flashing

To switch LED 1 on, the value -32768 (0x8000) must be entered in the address +2. However, the ASCII plain text protocol does not transfer this value.

SOLUTION:
Enter -32767 (0x8001) instead of -32768. This will set the flash bit of LED 8 which has no effect on the output.

16.5 Error Messages

Code	1	E_SLAVE_NOT_READY	Slave not ready
	2	E_PROTOKOL	Sequence of the packets
	3	E_FRAME	Protocol frame error
	4	E_TIMEOUT	Timeout error
	5	E_CRC_BCC	CRC error
	6	E_PARITY	Parity error
	7	E_SEND_ABORT	Send process aborted
	8	E_REC_ABORT	Receive process aborted
	9	E_BUF_SIZE	Insufficient cyclic buffer
	10	E_NO_DEFINE	No cyclic data defined
	12	E_DEFINE	Cyclic data already defined
	15	E_NO_PROTOCOL	Selected protocol is not supported
	16	E_OVERRUN	Receive buffer overrun
	40	E_SYS_ADDRESS	Undefined system variable
	50	E_HARDWARE	Communication error on the hardware level
	Subcode		
	1	E_HW_OCCU_OPEN_QUIT	Hardware error on acknowledging Occu Open
	2	E_HW_OCCU_CLOSE_QUIT	Hardware error on acknowledging Occu Close
	3	E_HW_REQ_QUIT	Hardware error on acknowledging Response
Code	51	E_PROTOKOL	Communication error on the protocol level
	Subcode		
	1	E_RESPONSE_NAK	Response acknowledged with NAK
	2	E_OCCU_CLOSE_NAK	Response acknowledged with NAK
	3	E_OCCU_OPEN_NAK	Response acknowledged with NAK
	10	E_TO_OCCU_OPEN_QUIT	Timeout error on acknowledging Occu Open
	11	E_TO_OCCU_CLOSE_QUIT	Timeout error on acknowledging Occu Close
	12	E_TO_REQ_QUIT	Timeout error on acknowledging Response
	20	E_RECEIVE_COUNT	Expected a different number of data bytes
	21	E_CHECKSUM	Echo and transmission do not match

Code 52 E_TELEGRAM . . . . . .Error in the telegram

Subcode	Error number from the controller	
1	0 + 01	Number exceeds limit
6	0 + 06	Wrong I/O code
7	0 + 07	I/O number out of range
10	10 + 0	CPU is available, however, no access authorization
24	20 + 4	Terminal has write authorization
26	20 + 6	Read authorization in use by another controller
28	20 + 8	Write authorization in use by another controller

INDEX

Symbole

A

B

C

D

E

F

G

I

L

M

P

R

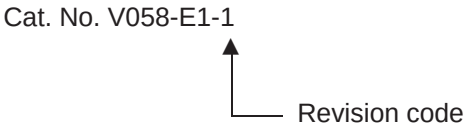
S

T

W

Revision History

A manual revision code appears as a suffix to the catalog number on the front cover of the manual.



The following table outlines the changes made to the manual during each revision. Page numbers refer to the previous version.

Revision code	Date	Revised content
1	March 1999	Original production

OMRON

Authorized Distributor:

Cat. No. V058-E1-1

Note: Specifications subject to change without notice.

Printed in the Netherlands

Regional Headquarters
OMRON EUROPE B.V.
Wegalaan 67-69
NL-2132 JD Hoofddorp
The Netherlands
Tel: +31 (0)23 56 81 300
Fax: +31 (0)23 56 81 388
Website: www.eu.omron.com